



TUGAS AKHIR - TE 145561

**PENGATURAN JARAK BACA UHF RFID BERBASIS VISUAL
BASIC .NET PADA *AUTOMATION TOOL'S CRIB***

Aldo Rahmad Ageng
NRP 10311500000028

Pembimbing
Dr.Ir. Ari Santoso, DEA.
Imam Arifin, ST., MT.
Ayyasy Az Zurqi, ST.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

***UHF RFID READ RANGE CONTROL BASED ON VISUAL
BASIC.NET IN AUTOMATION TOOL'S CRIB***

Aldo Rahmad Ageng
NRP 10311500000028

Supervisor
Dr.Ir. Ari Santoso, DEA.
Imam Arifin, ST., MT.
Ayyasy Az Zurqi, ST.

Automation Electronic Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2018

**PENGATURAN JARAK BACA UHF RFID BERBASIS
VISUAL BASIC .NET PADA AUTOMATION TOOL'S CRIB**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi**

**Institut Teknologi Sepuluh Nopember
LEMBAR PENGESAHAN**

Menyetujui,

Pembimbing I,

Pembimbing II,

Pembimbing III,

Dr.Ir. Ari Santoso, DEA

Dr. Ir. Imami Arifin, ST., MT.

Ayyasy Az Zurqi, ST.

NIP. 196602181991021001

NIP. 197302222002121001

NRP. 045

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

**PENGATURAN JARAK BACA UHF RFID BERBASIS
VISUAL BASIC .NET PADA AUTOMATION TOOL'S CRIB**

TUGAS AKHIR

Disusun oleh:

Aldo Rahmad Ageng

NRP.10311500000028

Menyetujui,

Deputy
Departemen *Machinery Production*,

Pembimbing Perusahaan,

Abu Hanifah, ST.
NRP. 081

Ayyasy Az Zurqi, ST.
NRP. 045

-----Halaman ini sengaja dikosongkan-----

PENGATURAN JARAK BACA UHF RFID BERBASIS VISUAL BASIC.NET PADA *AUTOMATION TOOL'S CRIB*

Aldo Rahmad Ageng
1031150000028

Pembimbing : Dr.Ir. Ari Santoso, DEA.
Imam Arifin, ST., MT.
Ayyasy Az Zurqi, ST.

ABSTRAK

Pengelolaan dan perawatan aset merupakan tindakan pencegahan dari kerusakan peralatan di industri. Tempat pengelolaan peralatan produksi atau yang biasa disebut *tool's crib* merupakan fasilitas bagi *field engineer* di *line* produksi. Secara umum, sistem pengelolaan alat yang beroperasi di industri dilakukan pengecekan dan perawatan setiap bulan. Namun, aktivitas pengecekan administrasi dan perawatan yang tidak seimbang berakibat terganggunya proses transaksi dan identifikasi. Otomatisasi pada *tool's crib* diterapkan untuk mengubah proses transaksi dan administrasi pada sistem menjadi otomatis menggunakan RFID (*Radio Frequency Identification*) dan database dalam proyek *Automation Tool's Crib*. RFID yang digunakan memiliki jarak baca yang terlalu jauh pada proses identifikasi. Oleh karena itu, diperlukan pengaturan jarak baca pada *Automation Tool's Crib* untuk mengurangi eror pada saat transaksi dan identifikasi. Pengaturan RFID berbasis Visual Basic.NET mampu mengurangi kesalahan proses identifikasi peralatan pada *Automation Tool's Crib* dari 7,5 % menjadi 1,2 %.

Kata kunci: *Otomasi, RFID, Tool's Crib,*

-----Halaman ini sengaja dikosongkan-----

UHF RFID READ RANGE CONTROL BASED ON VISUAL BASIC.NET IN AUTOMATION TOOL'S CRIB

Aldo Rahmad Ageng
10311500000028

Advisor : Dr.Ir. Ari Santoso, DEA.
Imam Arifin, ST., MT.
Ayyasy Az Zurqi, ST.

ABSTRACT

Asset maintenance is preventive action from tool's damage. Place for tool's production usually called by tool's crib, is a facility for field engineer on the production line. Generally, tool's maintaining system that operate in industry checked and maintain in every month. Nevertheless, administration check and maintenance are unbalanced that delayed transactional and identification process. Automating tool's crib applied to change transaction and administration activity on the system become automatic using RFID (Radio Frequency Identification) and database on Automation Tool's Crib project. Used RFID on this project have a read range that far too wide on the identification process. So, controlling read range of a RFID on Automation tool's Crib to reduce error on transaction and identification process. Controlling RFID based on Visual Basic.NET are able to reduce error of scanning process on Automation Tool's Crib from 7.5 % to 1.2 %.

Keywords: UHF RFID, Tool's crib, Automation.

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Allah SWT ang selalu melimpahkan rahmat dan rahim-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik.

Tugas akhir ini disusun untuk memenuhi persyaratan guna menyelesaikan Pendidikan Diploma tiga di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya yang berjudul:

PENGATURAN JARAK BACA UHF RFID BERBASIS VISUAL BASIC.NET PADA *AUTOMATION TOOL'S CRIB*

Penulis mempersembahkan Tugas Akhir ini kepada ayah dan ibu penulis yang memberikan berbagai bentuk ridho dan dukungan tanpa henti, Bapak Dr. Ir. Ari Santoso, DEA., Bapak Imam Arifin, ST., MT., dan Bapak Ayyasy Az Zurqi, ST. atas segala bimbingan ilmu, moral dan spiritual dari awal hingga Tugas Akhir ini selesai. Penulis juga mengucapkan terima kasih kepada kolega dan orang spesial yang membantu secara langsung dan tidak langsung pada saat proses pembuatan Tugas Akhir ini.

Penulis memohon maaf atas segala kekurangan pada tugas akhir ini. Demikian, Semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan teknologi industri di kemudian hari.

Surabaya, 2 Agustus 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan.....	4
1.5 Metodologi Penelitian.....	4
1.6 Sistematika Laporan	6
BAB II SISTEM AUTOMATION TOOL’S CRIB	9
2.1 <i>Tools Management</i>	9
2.2.1 <i>Tool Crib System</i>	10
2.2 <i>Automation Tool’s Crib</i>	12
2.3 RFID (<i>Radio Frequency Identification</i>)	14
2.3.1 Interrogator.....	14
2.3.2 Transponder.....	18
2.3.3 Komunikasi RFID	20
2.4 <i>Radiation Pattern</i>	21
2.5 Pengaturan Jarak Baca RFID.....	22
BAB III PERANCANGAN DAN IMPLEMENTASI	25

3.1	Perancangan Prosedur Sistem <i>Automation Tool's Crib</i>	27
3.2	Perancangan Spesifikasi Sistem <i>Automation Tool's Crib</i>	28
3.3	Perancangan <i>Hardware</i>	28
3.3.1	Rancang Bangun <i>Pilot Project Automation Tool's Crib</i> ..	29
3.3.2	Perancangan Sistem <i>Automation Tool's Crib</i>	30
3.3.3	Perancangan Rangkaian Daya <i>Automation Tool's Crib</i> ...	31
3.3.4	Perancangan Rangkaian I/O pada PLC	32
3.3.5	Perancangan Rangkaian Relay	33
3.3.6	Perancangan Posisi RFID <i>Interrogator</i> dan <i>Tag</i> RFID....	34
3.4	Perancangan <i>Software</i>	35
3.4.1	Perancangan Program Pengaturan Jarak Baca Otomatis..	37
3.4.2	Perancangan Program Pemilahan Tipe Data RFID	38
BAB IV PENGUJIAN DAN ANALISIS		41
4.1	Pengujian Komunikasi RFID	41
4.2	Pengujian Radiasi Jarak Baca RFID	43
4.3	Pengujian Pemindaian Peralatan Kerja	45
4.4	Pengujian Prosedur Sistem <i>Automation Tool's Crib</i>	47
4.5	Buku Manual <i>Automation Tool's Crib</i>	48
BAB V PENUTUP		51
DAFTAR PUSTAKA.....		53
LAMPIRAN		55
Lampiran 1. Program RFID		55
Lampiran 2. Hasil Pengujian Pemindaian Peralatan (30ms)		163
Lampiran 3. Hasil Pengujian Pemindaian Peralatan (50ms)		165
Lampiran 4. Manual Book <i>Automation Tool's Crib</i>		167
Lampiran 5. Datasheet RFID		180
Lampiran 6. Datasheet PLC CJ1M-CPU13		181
Lampiran 7. Datasheet Input Card CJ1M - ID211		183
Lampiran 8. Datasheet CJ1M – PA202.....		184
Lampiran 9. Datasheet Output Card CJ1M – OD211		185

Lampiran 10. Datasheet Ethernet Card CJ1M-ETN13	186
Lampiran 11. Datasheet LED indicator	187
Lampiran 12. Datasheet RFID TAG.....	188
RIWAYAT PENULIS.....	189

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

Gambar 2. 1	Diagram Manajemen Industri	9
Gambar 2. 2	Rangkaian transmitter radio.....	15
Gambar 2. 3	Diagram alur dari tipe tag class 1	18
Gambar 2. 4	Rangkaian I/Q (In-phase / Quadrature).....	19
Gambar 2. 5	Tag kartu RFID.....	20
Gambar 3. 1	Arsitektur Kerja Sistem Automation Tool's Crib... ..	25
Gambar 3. 2	Ruang Lingkup Kerja Automation Tool's Crib	26
Gambar 3. 3	Pilot Project Sistem Automation Tool's Crib	29
Gambar 3. 4	Tool's Crib di PT. Astra Otoparts Divisi WINTEQ	30
Gambar 3. 5	Desain Hardware Automation Tool's Crib	30
Gambar 3. 6	Rangkaian Daya Automation Tool's Crib	31
Gambar 3. 7	Rangkaian Input Di PLC Omron CJ1M.....	32
Gambar 3. 8	Rangkaian Output Di PLC Omron CJ1M.....	33
Gambar 3. 9	Rangkaian Relay untuk Solenoid Door Lock	34
Gambar 3. 10	Diagram Alur Sistem Automation Tool's Crib	35
Gambar 3. 11	Diagram Alur Komunikasi RFID.....	36
Gambar 3. 12	Diagram Alur Program Pengaturan Jarak Baca ...	38
Gambar 3. 13	Flowchart Pemilahan Data RFID	39
Gambar 4. 1	Pola Radiasi RFID UHF CT-I809.....	45

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

Tabel 4.1 Pengujian Komunikasi RFID	41
Tabel 4.2 Data Intensitas Daya.....	43
Tabel 4.3 Pemindaian dengan Interval Waktu 50ms	46
Tabel 4.4 Pemindaian Peralatan dengan Interval Waktu 30ms. ..	46
Tabel 4.5 Komponen Automation Tool's Crib.....	48

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mengenali serta menyelesaikan masalah di dunia kerja khususnya di industri di masa mahasiswa tentunya hanya diperoleh dengan cara magang. Berada di dunia kerja untuk beradaptasi dengan lingkungan industri tentu membiasakan mahasiswa untuk berpikir lebih luas dan tidak hanya berkutik dengan buku dan teori untuk berurusan dengan fenomena yang terjadi di industri. Masa magang dari mahasiswa yang menuntut mahasiswa untuk bekerja mengikuti ritme dari perusahaan tentu berbeda dari pengalaman dunia perkuliahan. Program magang di PT.Astra Otoparts Divisi WINTEQ mengantarkan penulis hingga melakukan penelitian yang diperuntukkan proyek akhir dari perkuliahan. Perusahaan yang bergerak di bidang manufaktur mesin industri, mengenalkan mahasiswa dengan berbagai penerapan instrument pada proses produksi di industri. Membangun mesin yang menyesuaikan kebutuhan dari *customer* menjadi tantangan bagi perusahaan untuk tetap mengikuti perkembangan industri di Indonesia. Pembuatan mesin di line produksi mengajarkan mahasiswa untuk bekerja sesuai manajemen waktu yang telah dipersiapkan secara detail disetiap pekerjaan yang telah diberikan kepada mahasiswa. Berada di line produksi dibawah departemen *machinery* dan *project management* mengenalkan penulis dengan kondisi bisnis dari perusahaan yang berwujud proyek di industri serta tahap pembangunan satu mesin dalam satu proyek. Pembangunan mesin yang ada dalam perusahaan dikerjakan dalam beberapa tahap besar yaitu tahap persiapan berupa konsep mesin, tahap desain mesin yang merupakan bagian *engineering* dan tahap *assembly* yang dikerjakan di *line* produksi oleh departemen *machinery*.

Pemahaman proses dari pembuatan mesin, memberikan tantangan bagi kami sebagai mahasiswa magang berinteraksi dengan industri dengan cara menyelesaikan permasalahan berwujud pembuatan suatu mesin yang membantu pekerjaan di line produksi. Fasilitas dan utilitas yang digunakan untuk produksi di industri manufaktur menentukan kualitas produksi dari perusahaan. Ikut andil dalam pembuatan mesin di industri membuat penulis membutuhkan peralatan yang mendukung kerja pada saat fase *assembly*, fasilitas yang meminjamkan peralatan

Peralatan kerja di *line* produksi pada industri manufaktur merupakan hal yang vital. Waktu ± 10 menit setiap hari terbuang karena kerusakan peralatan kerja tentu menjadi hambatan proses produksi dalam industri. Masalah penyediaan fasilitas kerja untuk *field engineer* perlu menjadi salah satu aspek yang selalu diawasi oleh industri. Mayoritas perusahaan jepang menerapkan sistem *kaizen* yaitu peningkatan kecil yang dilakukan terus menerus contohnya dalam pemanfaatan *Inventory management* di industri salah satunya pengelolaan peralatan kerja yang berada di *line* produksi, sehingga aset terorganisir dan diawasi dengan baik. Industri mengelola alat kerja sebagai layaknya aset karena pemeliharaan dan perawatan peralatan kerja mengurangi penambahan biaya untuk menunjang produksi[1]. Nilai hasil produksi yang dijaga dan dikembangkan dapat dipengaruhi oleh kondisi peralatan yang baik saat produksi.

Tempat penyimpanan peralatan kerja di industri atau yang biasa disebut *tool's crib* menjadi media yang baik untuk mengelola peralatan kerja[2]. Prosedur peminjaman dan pengembalian peralatan dilakukan dengan bantuan manusia untuk melayani *field engineer* di *line* produksi, namun prosedur yang diterapkan cukup memakan waktu dan pencatatan administrasi peminjaman serta pengembalian masih dilakukan secara manual, sehingga kurang efisien di sisi administrasi.

Menurut analisis hasil listing sensus ekonomi tahun 2016 oleh badan pusat statistik indonesia, industri manufaktur telah memberikan kontribusi terbesar terhadap perekonomian Indonesia terutama di sektor manufaktur mobil & sepeda motor sebesar (46,8%)[3]. Berdasarkan Survey Badan Pusat statisitik, tingkat kebutuhan penyediaan fasilitas untuk menunjang produksi contohnya, bahan baku, sumber daya manusia, dan fasilitas akan terus meningkat dan digunakan 75% dari jumlah industri. Penyediaan fasilitas di industri tentu menjadi bagian vital yang mempengaruhi laju produksi, sehingga pengelolaan fasilitas dalam industri menjadi poin penting dalam berjalannya bisnis.

Fasilitas pengelolaan alat kerja saat ini di industri, menyimpan lebih dari 100 buah alat yang digunakan setiap hari untuk keperluan manufaktur yang secara umum hingga tiga bulan tingkat daya tahan dan usia penggunaannya menurun. Peralatan kerja di sektor manufaktur adalah aset yang mempengaruhi kualitas dari hasil produksi disamping kemampuan dasar dari pekerja di bagian produksi. Sistem pengelolaan dengan model *tool's crib* merupakan metode pengelolaan dari peralatan kerja yang ada di industri, namun perbedaan standar prosedur dari metode

tersebut di setiap industri berbeda-beda, sehingga manfaat yang diberikan oleh metode ini tidak sama di setiap plan manufaktur[1]. Proses meminjam peralatan layaknya transaksi barter serta perawatan dari peralatan yang kurang intensif terhadap peralatan justru menjadi permasalahan dasar prosedur yang diterapkan.

Sistem pengelolaan peralatan menggunakan media *tool's crib* yang diperbarui, merubah proses transaksi menggunakan RFID untuk membantu administrasi dari peminjaman alat. Metode *real time monitoring* juga digunakan untuk mengawasi kondisi peralatan sebagai aset dari perusahaan. Teknologi *monitoring* pada *tool's crib* mengurangi beban kerja dari penjaga untuk mengolah data transaksi setiap hari sehingga untuk pelaporan peralatan sedikit lebih mudah dilakukan oleh admin[4].

Harapan dari Tugas Akhir ini akan dibuat mesin dan peralatan yang menunjang operasi dari *tool's crib* secara otomatis. Mesin ini menggunakan lemari penyimpanan yang dilengkapi dengan *solenoid lock* untuk mengunci lemari untuk menjaga keamanan dari peralatan. Serta menggunakan RFID jarak jauh sebagai pendeteksi peralatan yang ada dalam lemari ataupun diluar lemari. Sehingga peralatan yang dipinjam dan yang ada dilemari dapat dideteksi menggunakan pengendalian jarak baca dari RFID berdasarkan *database* yang mencatat transaksi dan informasi pokok dari peralatan serta pengguna secara otomatis.

1.2 Perumusan Masalah

Pengelolaan aset di industri tentu menerapkan metode tertentu untuk setiap aset yang dimiliki. Sistem pengelolaan aset produksi di PT. Astra Otoparts Divisi WINTEQ yang biasa disebut *tool's crib* kurang efektif untuk melacak peralatan dan pengelolaan administrasi dari transaksi peralatan yang menghabiskan ± 10 menit bagi *field Engineer* pada saat melakukan instalasi. Pekerjaan administrasi dalam *Tool's Crib* konvensional dilakerjakan pada saat peralatan selesai digunakan oleh pengguna untuk dicatat kondisinya. Oleh karena itu, beban kerja yang menumpuk hingga akhir bulan untuk administrasi harus dikurangi. Pembuatan mesin *Automation Tool's Crib* sebagai wujud otomatis dari sistem *tool's crib* mempermudah proses transaksi dan administrasi menggunakan database dan RFID sebagai metode pengenalan peralatan dan pengguna. RFID yang digunakan pada mesin tersebut masih memiliki kesalahan dalam tahap pelacakan alat dan pengguna yang perlu dipilah karena sering muncul gangguan data yang tidak tepat pada saat

pemindaian. Selain masalah data, juga terdapat kekuarangan akurasi dalam jarak baca RFID sehingga tidak sesuai pada tahap pelacakan peralatan serta *log-in*. oleh karena itu, fase pemindaian perlu disempurnakan agar dapat memenuhi kebutuhan pengguna di line produksi.

1.3 Batasan Masalah

Tugas akhir ini menggunakan RFID sebagai media pelacakan peralatan dalam *tool's crib* khususnya jenis UHF RFID dengan rentang frekuensi 908 – 928 Mhz untuk *reader* dan *transponder*. Peralatan yang dilacak oleh alat merupakan peralatan yang sering digunakan di line produksi serta memiliki luas penampang 36 cm² dengan menggunakan *tag* berbentuk *sticker* yang ditempelkan. Tipe komunikasi yang digunakan antara *reader* dan *transponder* menggunakan EPCC1-G2 yang merupakan protokol paling fleksibel dan terbaru untuk saat ini. PC sebagai kontroler dari RFID menggunakan Visual Basic.NET sebagai *platform* antarmuka untuk mengatur dan menerima data dari RFID menggunakan RS232 karena mampu diakuisi oleh *compiler*.

1.4 Tujuan

Pembuatan *Automation Tool's Crib* di PT. Astra Otoparts divisi WINTEQ untuk merintis pengelolaan peralatan kerja di line produksi yang sebelumnya konvensional menjadi digital, sehingga lebih efektif dan transaksi yang cepat. Penyempurnaan dari sistem tersebut salah satunya adalah meningkatkan akurasi baca RFID pada peralatan maupun identitas pengguna.

1.5 Metodologi Penelitian

Dalam pelaksanaan tugas akhir yang berupa pengaturan jarak baca UHF RFID berbasis Visual Basic.NET pada *Automation Tool's Crib*. Ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

1. Tahap persiapan

Mempelajari ilmu dasar dari pembuatan alat pada penelitian ini diperlukan untuk menentukan metode yang akan digunakan pada saat perancangan serta Analisa pengujian pada alat. Pemahaman dari penerapan *inventory management* di industri membantu memilih sistem yang menyesuaikan *tool's crib* yang

telah di terapkan. Proses identifikasi yang menjadi masalah pada pengelolaan inventaris memerlukan studi tentang RFID untuk memilih metode yang biasa digunakan pada identifikasi peralatan dan pengguna. Serta mempelajari karakteristik sinyal dari RFID untuk mendeteksi peralatan.

2. Tahap identifikasi dan permodelan sistem

Pada tahap ini identifikasi dari sistem alat sesuai data yang telah didapatkan dari studi literatur serta pemodelan dari alat yang akan dikerjakan.

3. Tahap perancangan

Pada tahap ketiga akan dilakukan sebuah perancangan dan pemodelan alat tersebut, mulai dari bagian sistem elektronik hingga sistem mekanik. Pada tahap ini, penggunaan RFID pada mesin *Automation Tool's Crib* digunakan untuk mendeteksi *tag* yang ada pada peralatan dan id card pengguna. alamat identitas tiap *tag* yang terbaca oleh *reader* dikirim melalui komunikasi RS232 untuk dikonfirmasi dengan *database*, sehingga dapat dibandingkan data yang tersedia dan terbaca.

Langkah selanjutnya setelah mendapatkan teori dasar sebagai dasar pembuatan tugas akhir ini, baru dilakukan perancangan tugas akhir ini. Perencanaan dilakukan agar alat yang dibuat memiliki hasil dan manfaat yang diharapkan. Perancangan yang dilakukan terdiri dari perancangan hardware yang meliputi rancangan rangkaian elektronika, mekanik dan software antarmuka .

4. Tahap pembuatan alat

Rancangan mekanik dan elektronika dibuat untuk memenuhi kebutuhan hardware. Komponen – komponen yang digunakan serta integrasi untuk membangun *Automation Tool's Crib* diperlukan integrasi antara HMI, *database*, RFID dan PLC sehingga prosedur yang direncanakan pada sistem sesuai. Perancangan rangkaian yang ada dalam panel dilakukan untuk memudahkan pada saat *troubleshooting* serta pembuatan program

antarmuka untuk mengakses data dari RFID dan dihubungkan dengan *database* sebagai data acuan.

5. Tahap pengujian dan analisa

Pada tahap ini, akan dilakukan kegiatan pengujian alat, menganalisa kesalahan pada alat dan mengatasi permasalahan tersebut, pada tahap ini, menganalisa parameter apa saja yang menjadi penyebab alat mengalami kesalahan. Tahap ini dilakukan pengujian meliputi pengujian komunikasi RFID untuk menentukan konfigurasi komunikasi paling ideal untuk sistem. lalu pengukuran radiasi jarak baca dari RFID yang menentukan cakupan wilayah yang mampu dijangkau oleh alat sehingga dapat diperkirakan transmisi daya dari RFID. Serta uji pindai dari RFID terhadap *tag* yang ada pada peralatan untuk mengukur seberapa sering terjadinya kesalahan pemindaian pada peralatan pada saat proses identifikasi dalam sistem.

6. Tahap penyusunan laporan

Setelah alat berhasil dibuat dan bekerja maka dilakukan pengambilan data dan analisa data untuk menentukan tindakan perbaikan dan evaluasi dari alat yang telah dibuat., dan tahap selanjutnya adalah penyusunan laporan untuk buku tugas akhir. Luaran dari buku tugas akhir ini bermanfaat bagi semua orang dan dapat dijadikan pedoman dalam melanjutkan dan mengembangkan ide tugas akhir ini.

1.6 Sistematika Laporan

Untuk pembahasan lebih lanjut, laporan tugas akhir ini disusun dengan sistematika sebagai berikut:

Bab I PENDAHULUAN

Membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan, sistematika laporan dan Metodologi tugas akhir yang dibuat

Bab II TEORI DASAR

Menjelaskan teori yang berisi teori – teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat

Bab III PERANCANGAN ALAT

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk pengoperasian alat.

Bab IV PENGUKURAN DAN ANALISA

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan analisa terhadap alat.

Bab V PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari tugas akhir ini dan saran – saran untuk pengembangan alat ini lebih lanjut.

-----Halaman ini sengaja dikosongkan-----

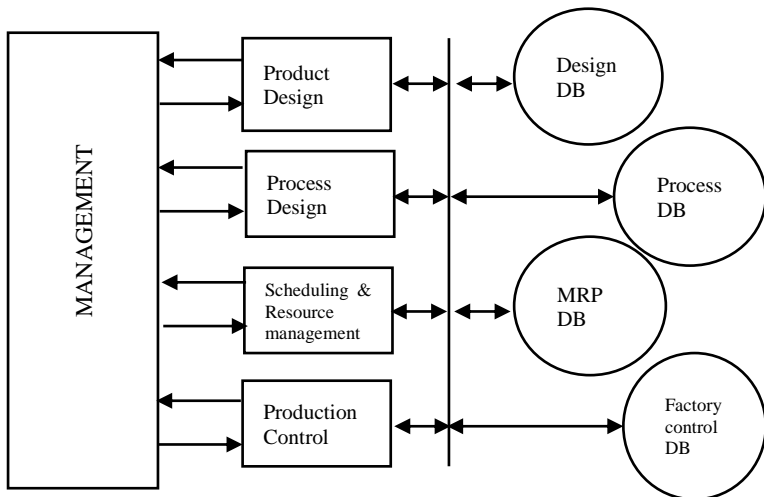
BAB II

SISTEM *AUTOMATION TOOL'S CRIB*

2.1 *Tools Management*

Sistem manajemen peralatan adalah usaha untuk mengintegrasikan berbagai fungsi peralatan termasuk pengawasan peralatan, transportasi, *inventory management*, dan *purchasing*. Untuk menentukan cakupan fungsi dari metode ini adalah mengetahui alur informasi yang ada pada sistem manufaktur yang telah terintegrasi dalam industry tersebut[5]. Alur aktivitas tersebut meliputi: *product design*, *process planning*, *scheduling and resource planning*, dan *production control*.

Informasi yang berhubungan dengan manajemen terbentuk dalam 2 bentuk. Yang pertama adalah perintah direksional yang digambarkan dengan gambar panah dari blok manajemen kearah ketiga fungsi utama.



Gambar 2.1 Diagram Manajemen Industri

Setiap fungsi melaporkan pekerjaan kembali ke manajemen untuk mendapatkan *feedback* yang digunakan oleh manajemen untuk mengukur performa dari unit secara objektif. Pada bagan diatas, kontrol produksi dibagi menjadi 3 sub-bagian, yaitu: *Factory Control*, *Workstation*

Control, dan *Machine Control*[2]. Pembagian sub diatas menunjukkan bahwa bagian dari fasilitas produksi dibagi menjadi kelompok yang berhubungan berdasarkan komunikasi / *control architecture* dari fasilitas produksi. Di dalam *lateral integration* dari control produksi, *scheduling*, perencanaan proses dan desain produk, pendukung dari sistem manufaktur dapat memanfaatkan *database* yang berhubungan dengan tiap dari bagian. Contohnya, bagian desain produk dapat membaca referensi desain yang ada di *database*. Perencanaan proses dapat mengakses fitur data dan geometris dari *database* CAD dan membuat perencanaan proses di *database* proses. Perencanaan sumber daya dan *scheduling* dapat mengakses perencanaan proses untuk menghasilkan capaian dari sistem produksi.

2.2.1 *Tool Crib System*

Sistem *tool crib* merupakan bagian dari manajemen peralatan yang bertujuan untuk meningkatkan produktivitas bagian produksi dan sebagai pengawasan dari kondisi peralatan selama pemakaian. Beberapa prosedur yang diberlakukan oleh sistem ini adalah:

a. *Inspection, recondition, dan disposal:*

Tool's crib mengawasi dan mengevaluasi peralatan yang digunakan di bagian produksi, contohnya peralatan standar, *drill*, dan *End Mill*[5]. Peralatan yang diawasi oleh sistem, dapat diberi tindakan langsung jika terjadi kerusakan atau memerlukan penggantian. Tindakan memperbaiki atau mengembalikan ke kondisi normal pada peralatan yang tidak habis pakai merupakan salah satu tugas dari sistem *tool's crib* yaitu *recondition*. Perbaikan dan evaluasi yang telah dilakukan oleh sistem akan berakhir pada tindakan *disposal* atau pembuangan peralatan yang disebabkan oleh peralatan yang tidak bisa digunakan kembali untuk produksi.

b. *Presetting*

Penyesuaian peralatan dengan pemakaian di *line* produksi menjadi tugas dari unit *tool's crib* untuk menyiapkan. Penggunaan peralatan yang membutuhkan konfigurasi tertentu untuk digunakan perlu dikalibrasi ulang setiap harinya[5] .

c. *Tools requisition unit*

Tahap lanjutan dari tindakan *recondition* dan *disposal* adalah tahap penggantian peralatan dengan yang baru. penyesuaian kebutuhan menjadi pertimbangan untuk menambahkan peralatan baru dengan adanya informasi berikut:

- *Recognized classification system*

Mempermudah pengelolaan dengan menyortir tipe, ukuran dan fungsi dari peralatan merupakan tujuan utama dari klasifikasi sistem. Penggunaan peralatan dengan identifikasi yang spesifik pada peralatan membuat petugas *tool's crib* lebih mudah mencari dan pengguna menggunakan alat yang lebih akurat secara fungsi.

- *Quantity of tools requirement*

Penyesuaian jumlah peralatan yang tersimpan di *tool's crib* harus memiliki jumlah minimal yang tidak menghambat operasi mesin untuk produksi[5], karena jumlah peralatan yang digunakan satu mesin dalam satu operasi tidak bisa digunakan bergantian dalam satu waktu. Menyediakan peralatan cadangan dengan jumlah yang cukup untuk aktivitas produksi lain secara bersamaan menjadi langkah yang tepat untuk membuat produksi tetap berjalan.

- *Availability of stock in the crib*

Jumlah peralatan yang cukup untuk melakukan aktivitas produksi, dengan menyiapkan cadangan untuk aktivitas produksi lain[5].

- *Quality of tools*

Baik dalam kualitas dan kondisi pada peralatan yang tersedia di *tool's crib*, meningkatkan nilai hasil produksi di akhir dari proses manufaktur. Pilihan peralatan dengan kualitas yang baik juga dipertimbangkan dengan pengalaman pemakaian dan nilai hasil produksi yang tinggi.

d. *Tool inventory*

Pengelolaan peralatan di penyimpanan *tool's crib* dalam jumlah banyak, mengharuskan sistem ini menyediakan informasi tentang :

- Penyediaan peralatan yang menyesuaikan aktivitas manufaktur di waktu yang akan datang.
- Pengelompokan peralatan berdasarkan penggunaan aktivitas produksi yang beragam, memudahkan pengguna untuk mengurangi kebutuhan peralatan tambahan untuk produksi
- Usia pemakaian dari peralatan yang harus tetap diawasi dan dirawat sehingga peralatan dapat digunakan dalam jangka waktu yang lama.
- Duplikasi persediaan peralatan yang ada dalam penyimpanan *tool's crib*, telah disesuaikan dengan jumlah mesin dan operasi yang akan berlangsung dalam satu hari, sehingga peralatan yang digunakan untuk produksi tersedia untuk semua mesin yang tersedia dengan waktu operasi yang bersamaan tanpa mengambat satu dengan yang lainnya.

e. *Tools performance evaluation unit*

Unit tugas evaluasi di *tool's crib* adalah bagian yang mengevaluasi kualitas dan performa peralatan yang ada di penyimpanan *tool's crib*. dokumentasi dari peralatan merupakan bentuk pengawasan dari *tool's crib* terhadap peralatan dan pengguna. Tindakan rekondisi dan pembuangan adalah tahap terakhir dari penilaian performa dan evaluasi dari peralatan.

2.2 *Automation Tool's Crib*

Tool's Crib konvensional yang diterapkan di industri manufaktur merupakan aset yang kurang diawasi bagian prosedur dan administrasinya. Peminjaman alat kerja oleh karyawan saat produksi meningkatkan kepadatan karyawan di line produksi saat pagi hari. Tingginya kebutuhan akan alat bantu ini memicu beberapa karyawan untuk membeli peralatan produksi dengan uang pribadi yang seharusnya adalah tanggung jawab dari perusahaan. Oleh karena itu, pengelolaan alat kerja di industri membutuhkan tindakan preventif untuk mengurangi terjadinya kepemilikan aset pribadi yang digunakan untuk produksi di industri. Pekerjaan administrasi yang dikerjakan oleh petugas *tool's crib* juga membutuhkan waktu 1 minggu untuk merekap transaksi alat dalam 1 bulan. Sistem otomatis dibuat untuk menggantikan proses administrasi

dan transaksi dari *tool's Crib* menggunakan RFID sebagai perangkat pengenalan dan pelacak, dan *database* untuk mencatat keseluruhan data karyawan, peralatan, transaksi hingga pengolahan data statistik peralatan.

Metode yang digunakan untuk mengenali karyawan yang ingin mengakses peralatan yang ada pada sistem menggunakan kartu akses yang termasuk dalam sistem RFID yang digunakan. Kartu akses yang



Gambar 2.2 *Cribmaster Accudrawer*

digunakan berisi data karyawan dan riwayat peminjaman dari pemilik kartu. Data dari pengguna yang ada dalam *database* menjadi kunci akses untuk melakukan peminjaman dan pengembalian dari peralatan kerja yang ada dalam *Tool's Crib*. Proses pelacakan alat yang diterapkan pada sistem memanfaatkan *tag* RFID layaknya kartu akses namun dalam bentuk yang berbeda. *Tag* ditempelkan pada peralatan kerja berfungsi sebagai pengenalan dan pelacak pada saat diluar tempat penyimpanan. Peralatan yang dipinjam oleh karyawan tercatat waktu dan identitas dari peminjam sehingga jika terjadi kehilangan dapat dipertanggung jawabkan data yang telah tercatat.

Transaksi peralatan pada *Tool's Crib* konvensional masih dilakukan oleh petugas penjaga. Oleh karena itu dapat diamati bahwa *tool's crib* tidak dapat beroperasi tanpa ada petugas, sehingga diluar hari

dan jam kerja normal, layanan ini tidak beroperasi bagi karyawan yang dalam waktu lembur dan dalam arti tidak fleksibel digunakan. Pengawasan transaksi menggunakan RFID pada sistem ini merubah fleksibilitas transaksi dari *tool's crib* menjadi dapat digunakan setiap waktu. Pelacakan peralatan yang telah dilengkapi dengan *tag* membantu proses transaksi peralatan dengan mengawasi secara langsung peralatan yang ada dalam tempat penyimpanan dengan menghitung jumlah dari peralatan secara berkelanjutan.

2.3 RFID (*Radio Frequency Identification*)

RFID adalah sistem yang secara jarak jauh dan nirkabel digunakan untuk mengidentifikasi peralatan lainnya yang disebut *transponder* atau yang biasa disebut dengan *tag* menggunakan *interrogator* atau *reader*[6]. Tag memiliki identitas unik yang biasa digunakan untuk mengidentifikasi objek yang ditandai.

Teknologi dari RFID dibagi menjadi 2 kategori utama, yaitu RFID aktif dan pasif. Teknologi RFID yang biasa digunakan saat ini adalah kategori pasif, karena transponder tidak memiliki sumber energi sendiri. Namun, transponder ditenagai oleh medan elektromagnetik yang dikeluarkan oleh *interrogator* yang mana dikonversi menjadi tenaga listrik. Pada teknologi transponder aktif, jarak kerjanya bisa melebihi kategori pasif, dikarenakan memiliki sumber tenaga sendiri [6].

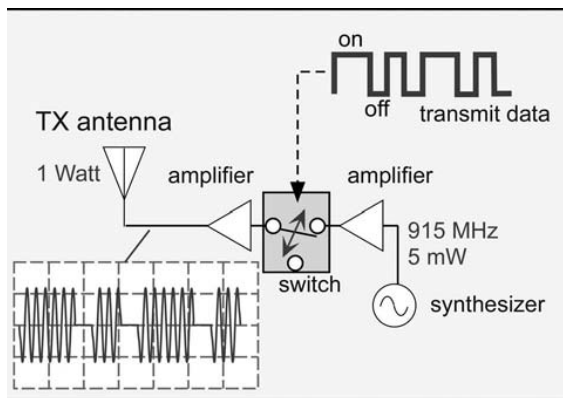
Berdasarkan frekuensi penggunaan dari *Interrogator*. RFID terbagi menjadi 3 bagian besar, yaitu LF (*Low Frequency*), HF (*High Frequency*), dan UHF (*Ultra High Frequency*). Perbedaan dari frekuensi pada *interrogator* berpengaruh pada jarak baca serta penggunaan dari RFID[6].

2.3.1 Interrogator

Transmitter RFID merupakan bagian dari sistem RFID yang digunakan untuk mengirim sinyal ke *transponder* atau penerima supaya data bisa diterima[6]. Tugas utama dari alat ini ada dua, yang pertama di fase *downlink* untuk menyediakan daya pada tag pasif dan modulasi sinyal dan di fase *uplink*, *interrogator* harus menyediakan sinyal yang tidak dimodulasi agar tag mampu mengembalikan data pada pengirim. *Interrogator* merupakan peralatan yang penting di sistem RFID, oleh karena itu harus dapat digunakan di banyak channel frekuensi radio secara akurat, dan dapat berpindah ke *channel* lainnya dengan cepat untuk

menyesuaikan dengan penggunaan. *Transmitter* yang baik dalam sistem RFID adalah yang memiliki jarak baca yang baik, sehingga untuk memenuhi persyaratan tersebut diperlukan daya yang besar yang mampu diterima oleh *tag*.

Rangkaian paling sederhana dari sebuah transmitter terdiri dari *synthesizer* sebagai penyedia sinyal pembawa yang digunakan sebagai media utama dalam pengiriman data pada radio[6]. Pengiriman data yang dilakukan oleh *transmitter* dengan menggunakan *synthesizer*, memiliki media pengiriman data yang menentukan frekuensi yang digunakan untuk berkomunikasi. Pengiriman data dengan media yang telah disediakan



Gambar 2.3 Rangkaian transmitter radio

dibutuhkan data untuk dikirimkan untuk berkomunikasi, pembuatan dan penentuan data yang akan dikirimkan dalam rangkaian *transmitter* adalah penggunaan *switch on/off* yang memberikan sinyal output digital dari sinyal yang dihasilkan oleh *synthesizer*. Penentuan *switch* dalam rangkaian transmitter berdasarkan jumlah data yang dikirimkan dalam satu kali komunikasi, semakin banyak data yang digunakan untuk berkomunikasi maka diperlukan saklar yang menghasilkan sinyal digital lebih cepat[7]. Transistor merupakan alat elektronik yang mampu digunakan sebagai saklar dalam rangkaian, kemampuan *switching* yang lebih cepat dari *switch* biasa menjadi penentu pengiriman data dari rangkaian *transmitter*. Sesuai dengan Gambar 2.3, sinyal output dari switch diperkuat menggunakan *amplifier* untuk menghasilkan output daya yang lebih besar dari sinyal *synthesizer*.

- **Efisiensi daya pengiriman**

Secara umum, konsumsi daya dari RFID *reader* hanya sebanyak 1W pada saat beroperasi, daya yang dikeluarkan untuk pengiriman data hingga menerima data kembali membutuhkan daya DC lebih besar untuk mencapai keluaran frekuensi radio 1 watt. Pemanfaatan amplifier dalam rangkaian *transmitter* untuk penguatan daya pengiriman, menjadi pengguna terbesar dalam rangkaian listrik dari *transmitter*, sehingga pemakaian *amplifier* yang kurang tepat dapat menimbulkan penggunaan daya yang kurang efisien. Penggunaan penguat secara efisien pada aplikasi penyediaan sinyal radio dapat dilakukan dengan memahami pengetahuan dasar dari penguat. Terdiri dari pembangkit arus dengan keluaran yang secara proporsional linear terhadap tegangan input, dimana sifat proporsional pada penguat yang konstan atau yang biasa disebut dengan *transconductance* dari penguat, membuat amplifier dapat menguatkan sinyal keluaran lebih besar daripada sinyal yang masuk.

- **Protokol komunikasi**

Di setiap proses dalam berkomunikasi seluruh kondisi dalam berkomunikasi tanpa sadar telah dilakukan, bentuk pesan yang kita sampaikan hingga arti dari pesan tersebut memiliki cara dan peraturan tersendiri untuk kedua belah pihak saling mengerti. Komunikasi dapat dikatakan berjalan dengan baik jika 4 keperluan dari komunikasi terpenuhi yaitu, media, bentuk pesan, akses dari media dan informasi. Di setiap dasar dari elemen protokol juga berlaku dalam sistem RFID khususnya UHF RFID. Penentuan protokol komunikasi untuk RFID dapat berpengaruh dalam perolehan informasi dari *tag / transponder*, serta kemampuan komunikasi yang berbeda di tiap protokol. Tag RFID dan RFID *reader* diharuskan untuk memiliki protokol komunikasi yang sama untuk dapat mengenali dan bertukar informasi. Penggunaan UHF RFID untuk jarak menengah memiliki protokol komunikasi yang dengan rentang frekuensi 902 – 928 MHz, penyesuaian frekuensi di antara kedua alat utama RFID harus dalam rentang frekuensi yang sama untuk dapat berkomunikasi.

Tag RFID pasif yang merupakan teknologi jarak pendek, memiliki beberapa permasalahan propagasi yang berhubungan dengan propagasi dalam ruang dan toleransi halangan. Kedua permasalahan tersebut timbul akibat cara kerja dari gelombang radio yang merambat dalam 3 cara, yaitu:

1. Perambatan secara langsung

Perambatan secara langsung dapat terjadi pada material atau bahan dielektrik seperti kertas, kardus, dan kayu. Radiasi gelombang radio pada material tersebut tidak mampu memantul dikarenakan memiliki daya pantul yang rendah dibandingkan dengan material lainnya.

2. Difraksi

Cahaya tampak memiliki panjang gelombang sekitar 0.5 *micron*. Perabotan manusia secara umum memiliki panjang gelombang ribuan bahkan jutaan untuk cahaya bisa menembus[6]. Namun, 900 MHz dari gelombang radio memiliki panjang gelombang sekitar 32 cm sehingga gelombang radio mampu dipancarkan dengan cara difraksi atau perbedaan panjang gelombang yang bervariasi di lingkungan.

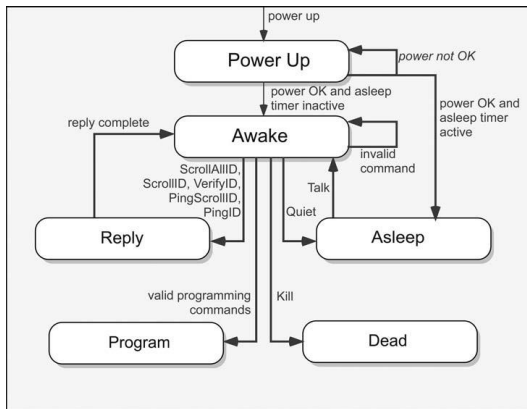
3. Refleksi

Berbagai macam benda dalam lingkungan rumah ataupun perkantoran seringkali dapat memantulkan gelombang radio. Benda bermaterial dielektrik seperti gelas dapat memantulkan gelombang radio secara efektif di sudut tertentu, air dan logam juga merupakan material pemantul yang baik untuk gelombang radio. Gelombang radio yang memantul pada objek tertentu dapat memantul ke titik buta dari pemancar, sehingga bagian yang tidak mampu digapai oleh *interrogator* dapat dipantulkan[6].

Protokol komunikasi ISO 18000-6C merupakan metode komunikasi untuk RFID yang paling sering digunakan, karena protokol ini mampu mengurangi gangguan dengan mengatur daya transmisi untuk komunikasi. Penggunaan protokol yang tepat pada sistem RFID dapat mempengaruhi akurasi dan kecepatan akuisisi data pada saat digunakan. Oleh karena itu, menentukan konfigurasi komunikasi pada perangkat akan membantu menentukan *preset* komunikasi yang dapat disesuaikan berdasarkan jarak kabel serial yang dicapai oleh pengirim. Memilih protokol *tag* RFID juga harus sama agar pengirim dan penerima dapat terdeteksi dan terhubung sesuai dengan peraturan pengiriman data yang ditentukan oleh protokol tersebut.

2.3.2 Transponder

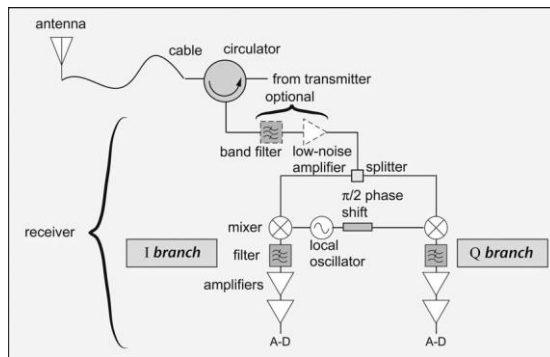
Sistem komunikasi pada RFID, untuk melengkapi sistem diperlukan perangkat yang memberikan umpan balik signal kepada *interrogator* saat mengirimkan signal informasi, *transponder* dapat menerima signal dari *interrogator* jika memiliki frekuensi, protokol dan orientasi yang sama dikedua alat komunikasi. Batasan dari tag pasif ada pada modulasi signal umpan balik yang memiliki frekuensi yang bervariasi, sehingga pada tag pasif, permasalahan demodulasi dan *decoding* jarang sekali terjadi, dibandingkan menggunakan komunikasi radio dengan teknologi berdasarkan pada signal fasa dan amplitudo dengan kode *error-correcting*. di sisi lain, tag pasif memiliki permasalahan dalam signal interferensi yang menghalangi signal dari



Gambar 2.4 Diagram alur dari tipe tag class 1

transmitter untuk dapat memberikan umpan balik, atau interferensi pada signal pantulan dari *tag*. Signal interferensi yang dipancarkan oleh *interrogator* yang memiliki amplitudo yang tidak beraturan dan diterima oleh tag akan mengalami efek dari perubahan modulasi sehingga saat kembali ke normal, *tag* dapat mengirimkan signal pantulan untuk mengirimkan kembali informasi. Penentuan frekuensi pada *tag* secara detail yang diatur pada *interrogator* berpengaruh pada pemilahan *tag* yang berbeda frekuensi dengan transmitter, sehingga komunikasi diantara dua perangkat dengan frekuensi yang sama dapat dikendalikan oleh *interrogator* sebagai faktor kontrol[4].

Dasar rangkaian sirkuit dari *tag* RFID adalah konversi secara langsung dari I/Q demodulator. Sinyal yang diterima oleh *transponder* akan dipisah dan diarahkan ke dua mixer yang satu untuk menyesuaikan dengan osilator dan yang lainnya digunakan untuk LO sinyal dengan pergeseran fasa sebesar 90 derajat. sinyal yang diterima dan diarahkan ke bagian *mixer* akan dicampur di tiap cabangnnnya dengan sinyal LO dan menghasilkan sinyal frekuensi rendah yang membawa respon dari *tag* dengan menghilangkan sinyal karier dan hamonik. Pada desain radio konvensional, sinyal yang diterima selalu diarahkan ke *band pass filter* saat pertama kali untuk menghilangkan sinyal interferensi dari luar pita yang dituju, dan untuk tag RFID UHF yaitu pada frekuensi 902-928 MHz. Penggunaan *band pass filter* yang secara umum pada radio digunakan, di

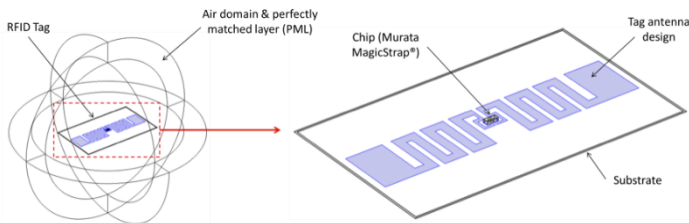


Gambar 2. 5 Rangkaian I/Q (*In-phase / Quadrature*)

RFID aplikasinya kurang membantu. Karena, kebutuhan sensitivitas dari radio untuk menghilangkan sinyal interferensi kurang dibutuhkan pada RFID, sehingga pemakaian pada tag pasif tidak selalu dibutuhkan, aplikasi LNA (*Low-Noise Amplifier*) pada radio konvensional digunakan untuk meningkatkan kekuatan sinyal lebih diutamakan dibandingkan *splitter* dan *mixer*[6]. transponder RFID umumnya tidak menggunakan aplikasi LNA karena metode pemantulan yang besar pada *tag* RFID saat masuk kedalam tag akan memasuki rentang level sinyal dari LNA yang dapat dikuatkan dengan linier secara baik, namun menyebabkan sensitivitas yang menurun pada sinyal tag, dan pada saat yang sama penguatan yang disediakan oleh LNA juga menguatkan fasa dan amplitude pada sinyal yang bocor dan mengakibatkan pengaruh yang

kurang baik daripada pemakaiannya di radio konvensional dengan adanya *noise* yang mendominasi sinyal pada *transponder*.

Penggunaan tag RFID, tidak hanya membutuhkan daya untuk bekerja, tetapi adanya arus DC sebagai sumber tegangan yang secara



Gambar 2.6 Tag kartu RFID

konstan sebesar 1-3V berdasarkan dari tipe transistor yang digunakan pada rangkaian, yang mampu menyediakan 10 *microampere* arus. Untuk mengubah arus AC menjadi DC, keperluan komponen elektrik yang merubah dan memperlakukan arus positif dan negatif secara berbeda yaitu dioda[8]. Komponen yang memperbolehkan arus untuk mengalir hanya ke satu arah, arus yang mengalir kearah yang diperbolehkan akan mengalir perlahan hingga mencapai tegangan yang diperbolehkan sehingga setelah dicapai akan meningkat secara cepat, dan sebaliknya pada arah yang berlawanan, aliran arus yang bocor meningkat saat tegangan juga meningkat[9]. Karena gelombang radio yang diterima oleh tag dalam bentuk arus AC maka penggunaan dioda pada IC *tag* diperlukan untuk mengubah arus yang digunakan pada *tag*. Dengan jumlah tegangan DC yang telah dikonversi dari gelombang radio ke arus DC, kebutuhan tegangan IC dari gelombang radio yang telah dikonversikan ke DC tidak mencukupi tegangan operasi dari IC.

2.3.3 Komunikasi RFID

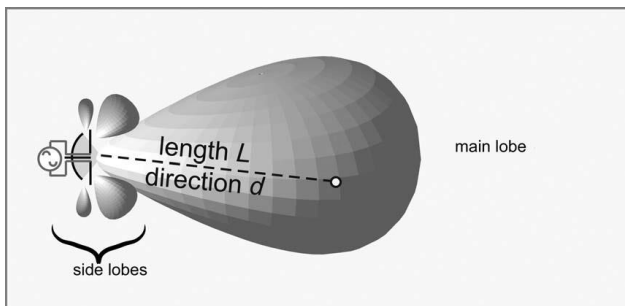
Komunikasi serial adalah metode komunikasi yang menggunakan satu bit data melalui satu kabel pada waktu tertentu. Dalam moda telekomunikasi, RS-232 merupakan standar yang direkomendasikan untuk komunikasi serial. Menentukan sinyal yang terkoneksi antara DTE (*Data Terminal Equipment*) dan DCE (*Data Communication Equipment*). RS232 biasa digunakan untuk *serial port* pada komputer. Standar komunikasi menentukan pula spesifikasi listrik, pola waktu sinyal dan

pinout connector[8]. Penggunaan komunikasi serial pada penelitian ini digunakan untuk berkomunikasi dengan RFID sebagai protokol komunikasi.

2.4 Radiation Pattern

Penggunaan titik ideal antenna isotropic pada RFID menghasilkan beberapa meter jarak baca menggunakan keluaran daya 1 Watt. Konsep antenna isotropik yang menjadi pedoman ideal yang mampu memancarkan sinyal ke segala arah secara merata, RFID memerlukan pemetaan arah dan wilayah pindai dari RFID sehingga pemanfaatan perangkat menjadi maksimal[6]. Transmisi daya yang dikeluarkan oleh alat harus disesuaikan dengan wilayah yang diinginkan, pemindaian diluar wilayah hanya akan membuang daya transmisi.

Kebutuhan antenna yang menyesuaikan permasalahan diatas mungkin diselesaikan menggunakan tipe antenna yang biasa dikenal dengan *directional antenna*. Direktivitas dari antenna dari peralatan tentu



Gambar 2.6 Direktivitas Antena

memiliki arah dan wilayah tertentu yang secara khusus aktif hanya dalam wilayah yang diarahkan. penentuan direktivitas dan wilayah dari pemancar khususnya RFID digambarkan dengan menunjukkan *antenna radiation pattern* yang dicontohkan pada Gambar 2.6. Jarak pada pola menggambarkan daya yang diradiasikan oleh antenna di arah tersebut. Arah dan wilayah maksimal yang digambarkan dalam pola merupakan direktivitas dari RFID, dan direktivitas yang digandakan oleh efisiensi radiasi adalah *Power Gain* dari antenna.

$$\Omega_{beam} = \frac{4\pi}{D_{Max}} \approx \frac{4\pi}{G} \quad (2.1)$$

Semakin tinggi penguatan dari *directional antenna*, maka semakin menyempit wilayah fokus dari energi yang diradiasikan. Pernyataan hubungan secara matematis dari energi yang dipancarkan dapat menggunakan pengukuran *solid angle*, yaitu rasio dari sudut total keliling bola(4π).

2.5 Pengaturan Jarak Baca RFID

Salah satu keunggulan dari UHF RFID adalah radiasi gelombang radio yang mampu menempuh jarak baca yang lebih jauh dibandingkan dengan LF(*Low Frequency*) dan HF(*High Frequency*) yang menggunakan induksi untuk transmisi data[7]–[12]. Radiasi gelombang radio yang dipancarkan oleh *reader* menyebar sesuai dengan direktivitas masing masing pemancar. Jarak baca dari RFID yang menyebar memiliki potensi untuk diatur dayanya sehingga direktivitas dan akurasi dari alat dapat ditentukan sesuai dengan kebutuhan[13]. Pengaturan jarak baca pada RFID dapat dipengaruhi oleh daya transmisi, posisi *tag*, orientasi *tag*, rentang frekuensi, dan interferensi yang merupakan parameter pada penelitian tugas akhir ini. Daya transmisi pada RFID berpengaruh pada kemampuan maksimum alat untuk mencapai jarak terjauh untuk membaca *tag*, dipengaruhi juga oleh antenna yang menjadi alat pemancar gelombang radio pada RFID menjadi variabel tetap untuk mengatur daya transmisi.

Daya transmisi dari RFID dipengaruhi oleh rangkaian amplifier yang terdapat pada RFID sebagai *gain* untuk meningkatkan daya. Penggunaan *amplifier* pada RFID terdiri dari lima parameter utama yang memberikan perubahan pada performa RFID *reader*[14].

1. *Gain*

Tujuan utama dari sebuah *amplifier* adalah untuk membuat sinyal *output* lebih besar dari sinyal *input*. oleh karena itu, parameter yang paling berpengaruh dalam pengaturan jarak baca adalah penyesuaian *gain* pada RFID dan *Amplifier* yang bekerja di frekuensi *microwave* secara umum menyediakan 10-20 dB dari *gain* yang merupakan faktor peningkatan 10 – 1000 dari daya *input*. Setiap kali sinyal dikonversikan ke rentang frekuensi kurang dari 1 MHz pada kebanyakan sistem RFID[6].

Pentingnya *gain* pada rangkaian RF karena adanya sinyal dari respon kemungkinan kecil.

$$\left\{ P_{out} = \frac{V_{out}^2}{2R} \right\} \rightarrow G = \frac{P_{out}}{P_{in}} = \frac{\frac{v_{out}^2}{2R}}{\frac{V_{in}^2}{2R}} = G_v^2 \quad (2.2)$$

2. *Power*

Parameter lain pada pengaturan jarak baca adalah kemampuan dari RFID untuk mencapai *maximum output power* yang biasa disebut dengan P_{sat} yang menunjukkan bahwa *output* dari *amplifier* telah tersaturasi dan tidak mampu bertambah lagi. Dalam perancangan RFID *reader*, daya keluaran rata – rata dari keluaran akhir pada *amplifier* menentukan daya maksimum yang mungkin disediakan oleh *reader*[6].

3. *Bandwidth*

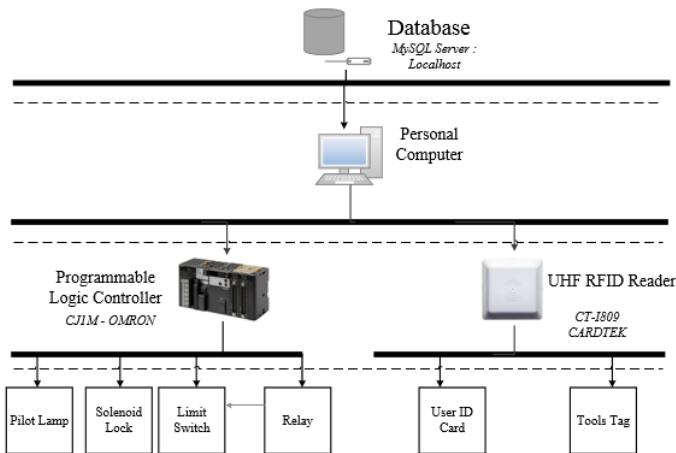
Pada umumnya transponder pada sistem RFID memiliki *gain* yang disesuaikan dengan kebutuhan dan membutuhkan rentang yang cukup. Penguat dari *transmitter* diharuskan dapat beroperasi secara penuh pada frekuensi UHF. Penting untuk dipahami bahwa saat rentang dari *amplifier* tepat dengan kebutuhan dari *reader* maka rentang yang berlebih dari nilai saat ini belum tentu sesuai dengan kebutuhan dari *reader*[8].

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN *AUTOMATION TOOL'S CRIB*

Tahapan dari perancangan dan pembuatan penelitian yang berjudul Pengaturan Jarak Baca RFID *Interrogator* pada Mesin *Automation Tool's Crib* di PT. Astra Otoparts Divisi WINTEQ, akan dibahas secara bertahap. Pembuatan Mesin *Automation Tool's Crib* terdiri dari pembuatan *pilot project* dari sistem, perancangan komunikasi RFID, Perancangan rangkaian daya untuk panel, perancangan rangkaian Input-Output pada PLC, perancangan rangkaian *relay* untuk *solenoid lock*,

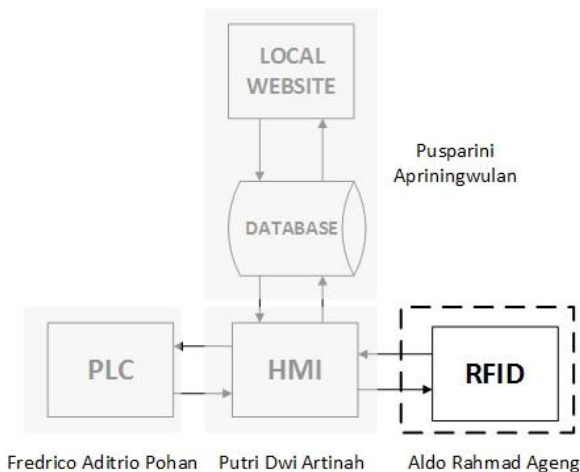


Gambar 3. 1 Arsitektur Kerja Sistem Automation Tool's Crib

perancangan komunikasi PLC, perancangan posisi *tag* RFID, perancangan komunikasi *database*, perancangan desain HMI menggunakan Visual Basic.NET, dan perancangan pengaturan jarak baca otomatis pada RFID.

Berikut adalah penjelasan mengenai diagram Alur pada Gambar 3.1, Penggunaan PC / Laptop digunakan sebagai kontroler sebagai *Master* untuk RFID dan PLC. RFID pada mesin digunakan untuk mendeteksi tag berupa kartu identitas pengguna dan identitas peralatan. *Tag* RFID yang digunakan untuk pengguna berupa kartu dipakai untuk mengakses

layanan dari mesin, dan *tag* pada peralatan digunakan untuk mengidentifikasi peralatan yang akan dipinjam atau dikembalikan oleh pengguna pada sistem, data yang dibaca oleh *RFID interrogator* dikirimkan melalui komunikasi serial yang terhubung menggunakan kabel serial tipe RS232 yang dikonversikan ke USB (*Universal Serial Bus*) yang dipasang pada PC, lalu data diakuisi oleh program aplikasi windows menggunakan platform Visual Basic.NET sebagai penerima data serial. PLC sebagai *local controller* pada mesin khususnya *LED indicator*, *solenoid lock*, *keyswitch*, *limit switch*, dan *buzzer*. Komunikasi PLC OMRON dengan PC menggunakan OPC CX-Serverlite yang diprogram melalui Visual Studio, sehingga input dan output dari PLC



Gambar 3.2 Ruang Lingkup Kerja *Automation Tool's Crib*

dapat dimonitoring dan dikendalikan melalui PC. Sistem akan mengunci akses pada sistem yang berbentuk lemari menggunakan *solenoid lock* yang dikendalikan oleh PLC. Rangkaian solenoid lock membutuhkan *relay* dikarenakan tegangan dan arus yang diperlukan untuk beroperasi tidak sesuai dengan tegangan *output* dari PLC. PC sebagai master dan juga HMI melakukan integrasi terhadap RFID, dan PLC yang diimplementasikan pada sistem *automation tool's crib*. Fungsi database pada sistem ini digunakan untuk menyimpan data dari peralatan, pengguna dan *IDnumber* dari *tag* RFID sehingga dapat disesuaikan

kebutuhan dengan sistem. Pembuatan layanan administrasi otomatis juga terdapat dalam sistem untuk mencatat aktivitas yang telah dilakukan oleh user pada saat menggunakan sistem yang akan dimasukkan ke dalam *database*.

3.1 Perancangan Prosedur Sistem *Automation Tool's Crib*

Sistem konvensional dari *tool's crib* memiliki 2 prosedur yang ditetapkan untuk meminjam peralatan yang ada dalam penyimpanan. Prosedur transaksi yang terdiri dari peminjaman dan pengembalian serta identifikasi yang digunakan untuk mengenali peralatan dan pengguna diterapkan untuk menjaga keamanan dari *tool's crib*. Karyawan di *line* produksi sebagai pengguna, memiliki koin sebagai batasan peminjaman alat dan pengenalan untuk menunjukkan identitas peminjam. Menggunakan koin pada transaksi sebagai pengenalan yang diletakkan pada tempat peralatan yang dipinjam menambah waktu pencarian alat pada saat akan dikembalikan atau pada saat ada kehilangan alat. Data peminjaman dari alat menggunakan kombinasi *database* dan RFID memungkinkan sistem untuk melakukan *tracing* apabila terjadi kerusakan atau hilangnya peralatan tanpa harus mengecek peralatan[2].

Identifikasi pada sistem otomatis menggantikan koin dengan kartu RFID sebagai pengenalan untuk mengakses sistem dengan membuka *solenoid door lock* pada lemari, sehingga peralatan yang ada dapat dipinjam atau alat yang telah dipinjam dapat dikembalikan. Pengguna harus mendekatkan kartu RFID didepan muka lemari pada jarak ± 50 cm untuk kartu dapat terbaca oleh *RFID reader*. Pada proses identifikasi, RFID mendeteksi adanya tag terdekat secara terus menerus dan membandingkan data yang diterima dengan *database* sebagai acuan data. Akses yang telah diberikan kepada pengguna memicu PLC untuk membuka kunci dengan *solenoid door lock* dan memberikan indikator pada HMI dan *pilot lamp* untuk menampilkan data dari pengguna. Untuk alasan keamanan, pengunci lemari dikunci kembali apabila pengguna tidak membuka pintu lemari selama 10 detik dan apabila pengguna telah membuka pintu lemari, *limit switch* mendeteksi pintu yang terbuka sebagai picuan dari RFID untuk berhenti melakukan pemindaian dikarenakan adanya perubahan data apabila ada aktivitas peminjaman dan pengembalian. Proses pemindaian RFID untuk menghitung dan identifikasi peralatan dilakukan pada saat sebelum dan sesudah pengguna membuka pintu dari lemari. Sehingga dari perbandingan jumlah data

pindai pada RFID jika ada perbedaan data maka data yang berbeda adalah peralatan yang dipinjam atau dikembalikan kedalam sistem.

3.2 Perancangan Spesifikasi Sistem *Automation Tool's Crib*

Pembuatan sistem otomatis yang digunakan untuk memperbarui sistem konvensional memiliki aspek yang harus terpenuhi sehingga menampakkan perubahan dari sistem. Aspek yang menjadi pertimbangan merupakan kekurangan dari sistem *tool's crib* konvensional. Oleh karena itu, kebutuhan sistem yang melatarbelakangi permasalahan dari *tool's crib* menjadi persyaratan dan target dari perancangan sistem otomatisasi dari *tool's crib*.

Karyawan yang bertugas untuk menjaga *tool's crib* merupakan karyawan yang juga melayani pengguna untuk meminjam peralatan yang ada di penyimpanan, dan juga mencatat riwayat peminjaman dari pengguna. Tugas bagi karyawan untuk melayani dan mengelola administrasi pada *tool's crib* kurang efektif untuk melayani *engineer*. Oleh karena itu, pelayanan otomatis mulai dari identifikasi, transaksi, dan administrasi dari *tool's crib* diharapkan mampu menyelesaikan masalah. Antrian karyawan di pagi hari untuk meminjam peralatan produksi di *tool's crib* membuang waktu efektif dari produksi di industri. Proses transaksi yang terhitung ± 2 menit lamanya dan administrasi yang tertunda menambah *time cost* dalam produksi di industri manufaktur. Sistem otomatis yang direncanakan dibuat mampu bekerja secara otomatis dan membutuhkan administrator untuk mengecek dan *maintenance* data dari peralatan yang telah tersimpan di database sebagai data acuan dalam sistem. Proses administrasi yang sebelumnya ditulis secara manual di buku harian menjadi diharuskan menjadi bentuk laporan bulanan yang menambah beban pekerjaan bagi petugas *tool's crib*.

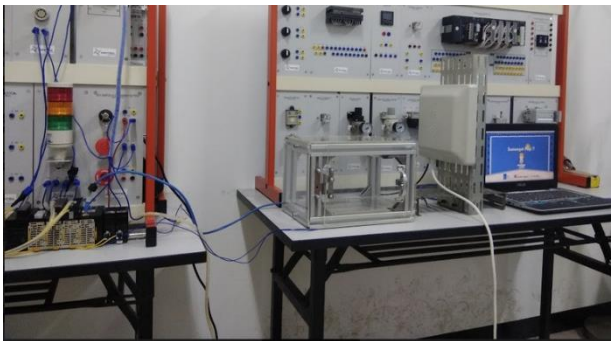
3.3 Perancangan *Hardware*

Bagian perangkat yang telah dirancang dibagi menjadi sub bab yang dijelaskan secara per sub bab nya, meliputi:

1. Perancangan *pilot project* dari sistem *Automation tool's Crib*
2. Perancangan sistem *Automation Tool's Crib*
3. Rangkaian daya pada panel mesin *Automation Tool's Crib*
4. Rangkaian *input* dan *output* pada PLC Omron CJ1W
5. Rangkaian *Relay* untuk *Solenoid Door Lock*
6. Perancangan Posisi RFID *Interrogator* dan *Transponder*

3.3.1 Rancang Bangun *Pilot Project Automation Tool's Crib*

Pilot project merupakan implementasi sistem dalam skala yang lebih kecil dari yang digunakan untuk simulasi, sehingga ditemukan masalah dan halangan yang akan terjadi jika diwujudkan dalam bentuk sistem yang sebenarnya. Solusi dan pencegahan yang diterapkan pada masalah yang muncul pada saat simulasi akan memudahkan pembuatan sistem skala besar. Arsitektur dan diagram alur sistem *Automation Tool's Crib* akan diterapkan dengan menggunakan lemari dan RFID tag yang lebih sedikit. Simulasi sistem peminjaman dan pengembalian dapat



Gambar 3. 3 *Pilot Project Sistem Automation Tool's Crib*

dilakukan dengan menempelkan kartu RFID pengguna mendekati wilayah baca RFID, lalu HMI akan menampilkan layar identitas pengguna, dan pengguna dapat meminjam peralatan yang ada di dalam lemari. Setelah transaksi, peralatan yang dipinjam tercatat dalam *database* sebagai riwayat transaksi.

3.3.2 Perancangan Sistem *Automation Tool's Crib*

Sistem *Tool's Crib* merupakan sistem layanan penyediaan peralatan untuk insinyur lapangan yang ada di PT. Astra Otoparts Divisi WINTEQ. Pelayanan dari *tool's crib* memerlukan bantuan di bagian administrasi dan penjaga untuk melayani peminjaman serta mencatat peminjaman dari pengguna, sehingga kebutuhan sistem yang otomatis lebih tepat digunakan untuk *tool's crib*. Sistem pelayanan *tool's crib* konvensional dilakukan menggunakan koin yang tertulis inisial nama pengguna yang berfungsi untuk melacak dan sebagai jaminan saat digunakan. Kepemilikan koin setiap orang berjumlah 5 - 10 koin setiap orangnya, sehingga peminjam dapat meminjam peralatan sejumlah



Gambar 3.4 Desain *Hardware Automation Tool's Crib*

dengan koin yang dimiliki. Sistem otomatisasi pada *tool's crib* yang direncanakan, memiliki kemampuan untuk mencatat administrasi saat peminjaman dan pengembalian barang, sehingga pekerjaan administrasi

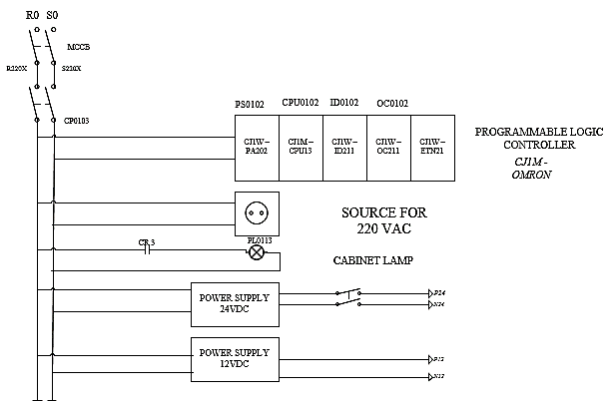


Gambar 3.5 *Tool's Crib* di PT. Astra Otoparts Divisi WINTEQ

dapat diatasi dengan sistem[5]. Lalu mampu menggantikan tenaga penjaga *tool's crib* karena peminjam melakukan *self service* saat transaksi barang. Sistem *Automation tool's crib* memiliki sistem keamanan yang menjaga peralatan yang ada di dalam lemari dengan lemari yang telah dimodifikasi dengan ditambahkan *solenoid door lock* yang mampu menutup dan membuka kunci pintu secara otomatis.

3.3.3 Perancangan Rangkaian Daya *Automation Tool's Crib*

Distribusi daya pada panel mesin *Automation tool's crib* dirancang untuk memberikan suplai tegangan ke seluruh komponen yang ada dalam panel, meliputi: PLC Omron CJ1M, 24VDC *power supply*, 12VDC *Power Supply*, dan relay. Rangkaian yang ada dalam panel, membutuhkan tegangan sumber sebesar 220V, pengamanan tegangan panel menggunakan MCB (*Magnetic Circuit Breaker*) yang memiliki *breaking capacity* sebesar 2.5 KA sehingga jika terjadi peningkatan daya diluar batas *breaking capacity*, maka MCB akan memutus aliran suplai. Tegangan yang dialirkan di komponen diatas memerlukan *circuit protector* yang khusus melindungi komponen yang alirannya terputus oleh *circuit protector*. Dengan *breaking capacity* sebanyak 2A, komponen

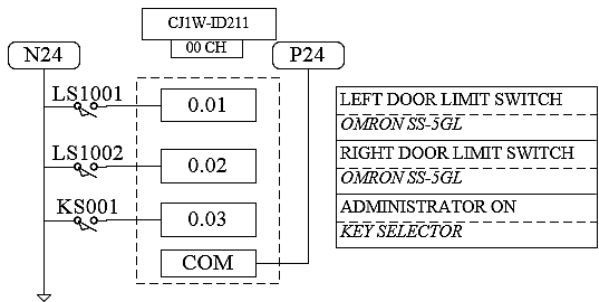


Gambar 3. 6 Rangkaian Daya *Automation Tool's Crib*

ini akan memutus rangkaian yang jika terjadi keadaan dimana besar arus melewati batas maksimal dari *circuit protector*. PLC membutuhkan sumber tegangan 220V dan 6.2 mA untuk beroperasi.

3.3.4 Perancangan Rangkaian I/O pada PLC

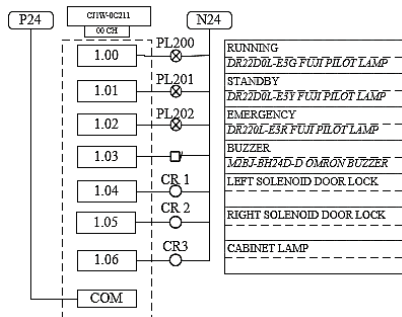
PLC sebagai kontroler tingkat area lokal memiliki Sensor dan aktuator yang harus dikendalikan untuk memenuhi kebutuhan sistem. *Automation Tool's Crib* merupakan mesin yang menggunakan PLC sebagai kontroler untuk melakukan layanan secara otomatis yang dikendalikan oleh PLC. Penggunaan PLC dengan tipe modular memudahkan pengguna untuk menyesuaikan ukuran dan tempat yang tersedia dalam panel, dan penyesuaian kebutuhan dari pengguna. Penggunaan PLC modular Omron tipe CJ1M dengan pengaturan common NPN yang ditambahkan dengan modul *input* tipe CJ1W – ID211 yang



Gambar 3. 7 Rangkaian Input Di PLC Omron CJ1M

memiliki 16 *points* input untuk membaca sensor atau komponen input lainnya. Komponen yang digunakan untuk input pada automation tool's crib adalah limit switch yang dipasang pada pintu lemari, untuk mendeteksi kondisi pintu terbuka atau tertutup. Modul output yang dipasangkan pada PLC CJ1M yaitu tipe CJ1W – OC211 yang memiliki 16 *points* output dengan common NPN untuk mengaktifkan indikator dan aktuator yang ada pada sistem. Komponen output yang digunakan pada sistem terdapat LED indicator, solenoid door lock, buzzer, dan relay. Indicator yang terpasang di sistem digunakan untuk menunjukkan kondisi dari sistem, jika indicator menyala merah berarti terjadi suatu aktivitas yang menyalahi prosedur / kesalahan dari sistem disertai suara alarm dari buzzer, lalu jika menyala kuning berarti sistem berada dalam kondisi standby sehingga siap digunakan, dan jika indicator menyala hijau, maka

sistem sedang digunakan dan beroperasi. *Solenoid door lock* digunakan untuk mengunci dan membuka pintu lemari saat pengguna akan



Gambar 3. 8 Rangkaian Output Di PLC Omron CJ1M

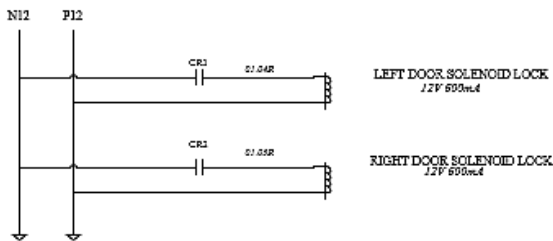
mengakses sistem. Jika mesin sedang tidak digunakan, pintu harus dalam keadaan tertutup dan terkunci.

3.3.5 Perancangan Rangkaian Relay

Keamanan sistem dari *automation tool's crib* terletak pada titik yang paling mudah untuk pencurian / menyalahi prosedur yaitu pada pintu lemari pada saat proses transaksi. Peralatan yang dijaga dalam lemari tercatat dalam administrasi untuk data peralatan yang tersedia dalam lemari. Lemari yang terkunci secara sistem memiliki keuntungan dalam sisi keamanan dan prosedur. Sisi keamanan dari sistem mencegah adanya transaksi ilegal yang kurang menguntungkan bagi pengguna, dan dengan prosedur penguncian pintu secara otomatis, maka peminjaman peralatan akan berjalan sesuai prosedur sistem yang telah direncanakan. Pemasangan *solenoid door lock* pada pintu lemari akan mengunci secara otomatis saat pintu lemari ditutup, dan akan membuka kunci saat pengguna sudah terdeteksi sebagai user pada saat itu. Rangkaian relay dibutuhkan untuk mengaktifkan *solenoid door lock* yang memiliki tegangan operasional 12V 2A, sedangkan pada modul output PLC memiliki tegangan output sebesar 24V, sehingga melewati tegangan operasi maksimum dari *solenoid door lock*. Penggunaan *relay* tepat

digunakan untuk fungsi *switching* yang menghubungkan 2 tegangan berbeda dalam 1 output.

Tegangan sumber untuk mengaktifkan *coil relay* menyesuaikan dengan tegangan output dari PLC yang menggunakan common NPN, dan untuk contact dari *relay* menggunakan kontak *normally open* dengan menghubungkan sumber tegangan negative dari power supply 12V dengan polar negative dari *solenoid door lock*. Pada saat output dari PLC memberikan tegangan 24V kedalam *coil relay*, maka kontak yang



Gambar 3. 9 Rangkaian Relay untuk *Solenoid Door Lock*

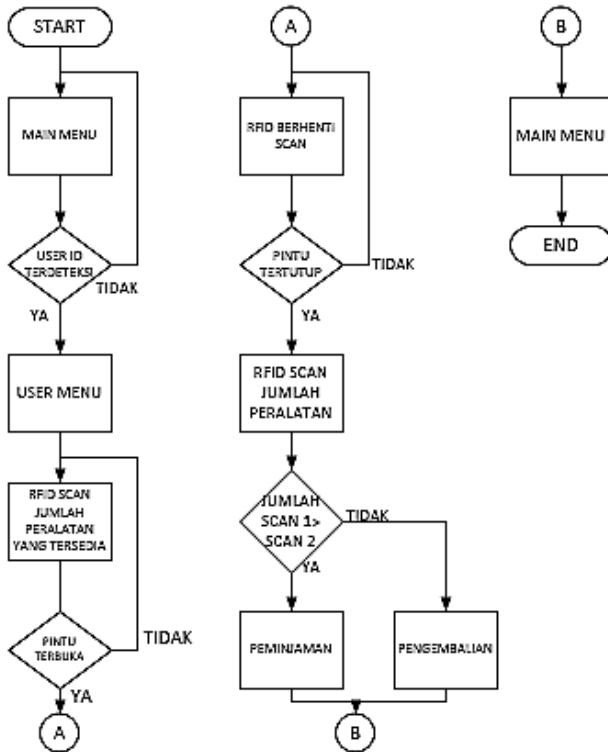
mulanya *Normally Open* menjadi tersambung dan medan magnet yang terbuat dari lilitan *solenoid* menarik kunci sehingga lemari bisa dibuka dan pengguna dapat mengambil peralatan yang dibutuhkan.

3.3.6 Perancangan Posisi RFID *Interrogator* dan *Tag* RFID

Penentuan posisi dari sistem peralatan RFID dipengaruhi oleh propagasi dan orientasi. Penempatan *interrogator* sesuai dengan kemampuan baca, meningkatkan efektivitas dari proses baca tag. Mengukur jarak baca dari *interrogator* menggunakan metode radio radiation pattern dapat diketahui sudut elevasi dan azimuth dari RFID. Penempatan di atas dari lemari dengan posisi *tag* yang berhadapan dengan reader merupakan posisi yang paling efektif untuk membaca tag yang telah terpasang pada peralatan tersebut. Penempatan dari RFID menentukan kemampuan baca dari RFID yang dipengaruhi oleh material sekitar dan juga jarak dari *tag* yang telah ditempelkan pada peralatan. Oleh karena itu, posisi dari UHF RFID diletakkan di atas lemari penyimpanan sehingga orientasi dari RFID tidak terpengaruh oleh kemiringan *tag* yang ditempel pada peralatan, tetapi pada *tag* yang jauh.

3.4 Perancangan Software

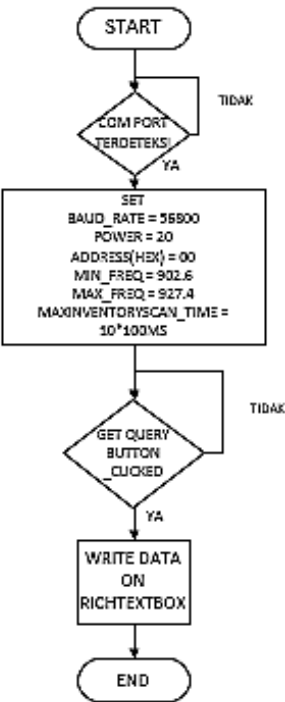
Perancangan software dan komunikasi dari alat ini menggunakan .NET sebagai platform pemrograman software Microsoft Visual Studio untuk *compiling* dan *interface*. Program yang dibuat untuk alat ini meliputi komunikasi dan sistem dari alat itu sendiri. Sistem ini diawali



Gambar 3. 10 Diagram Alur Sistem *Automation Tool's Crib*

dengan sekuens tampilan pada PC dan *scanning tag* RFID yang berbentuk kartu dibelakang layar sesuai dengan Gambar 3.8. Data yang dibaca oleh RFID berbentuk data serial yang selanjutnya dibandingkan dengan *database* pada platform Microsoft Access, jika data yang dibandingkan sama dengan yang ada di *database*, maka pengguna yang memiliki kartu

akses dapat menggunakan alat. Data serial yang telah terdaftar di *database* akan menampilkan data dari pengguna dan program memberikan perintah kepada PLC untuk menyalakan indikator dan membuka kunci *solenoid door lock*, dan RFID membaca *tag* yang ada



Gambar 3. 11 Diagram Alur Komunikasi RFID

pada peralatan untuk mengetahui peralatan yang tersedia dalam lemari. Jumlah barang yang terdeteksi pada *scanning* pertama akan dimasukkan pada tabel sebagai *buffer data* yang dikirimkan dari RFID. Data yang dikirimkan oleh RFID akan terus menerus dikirimkan hingga pengguna membuka pintu lemari sebagai pemicu untuk menghentikan dan menyudahi scanning data yang pertama. Pintu lemari yang terbuka memberikan akses kepada pengguna mengambil peralatan yang akan digunakan. Pintu yang ditutup oleh pengguna, menandakan akhir dari

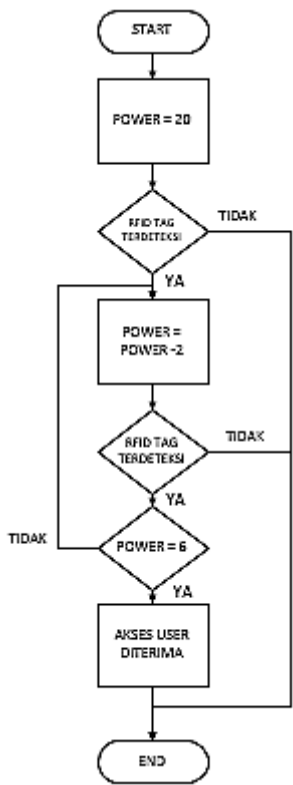
proses pengambilan dan memulai RFID untuk proses *scanning* yang kedua. jumlah peralatan dari *scan* pertama dan kedua, dibandingkan dan jika jumlah peralatan yang terdeteksi pertama lebih besar dari yang kedua maka akan dihitung sebagai peminjaman alat, dan sebaliknya sebagai pengembalian alat. Proses peminjaman dan pengembalian alat merupakan proses olah data dengan database, menulis dan menghapus data sesuai dengan transaksi yang telah dilakukan. Setelah pengguna menutup pintu lemari dari alat, pintu akan terkunci secara mekanis sehingga pengguna harus memasuki akses ulang jika ingin membuka pintu kembali.

Pengiriman data serial dari RFID memerlukan konektivitas serial yang ada pada PC dengan memanfaatkan komponen serial COM port. Media dapat diakses dari Visual Studio dengan *menambahkan tools serialComPort* untuk mengakses COM yang tersambung dengan USB. Program yang dibuat untuk menghubungkan dan memutuskan sambungan data serial memerlukan *library* dari RFID yang digunakan untuk mengenali program konektivitas yang dibuat di Visual Studio. Sesuai dengan Gambar 3.11, komunikasi dari RFID dapat dibuka dengan mengenali port COM yang sedang tersambung dengan perangkat, koneksi dari port yang telah terbuka memerlukan deklarasi penentuan alamat awal data, *baudrate*, rentang frekuensi, daya pancar, kapasitas baca dan kecepatan baca perangkat. Setelah parameter yang diperlukan sudah ditentukan, data akan terbaca pada serial monitor yang tertera pada tampilan.

3.4.1 Perancangan Program Pengaturan Jarak Baca Otomatis

Daya pancar dari RFID dengan frekuensi UHF mampu mencapai hingga 4 m. lemari yang memiliki 1,5 m memerlukan pengaturan jarak baca agar tag yang terbaca lebih efektif dan bagian luar lemari tidak terbaca. Diluar parameter dari pengaturan daya, tag RFID juga menyesuaikan orientasi dan posisi dengan *reader*. Pengaturan daya yang tepat akan menentukan validitas data yang dibaca pada saat *scanning* peralatan. Pengaturan dilakukan pada saat *scanning* kartu pengguna di awal prosedur serta di saat *scanning* peralatan untuk yang pertama maupun yang kedua. Proses *scanning* pengguna memakai daya pancar sebesar 20 dB yang berubah secara dekremen untuk memastikan bahwa user dalam posisi jangkauan terdekat untuk menggunakan layanan dan

dalam proses *scanning* daya akan berubah menjadi 10 dB dan akan meningkatkan kapasitas baca tag per ms sesuai pada Gambar 3.10.

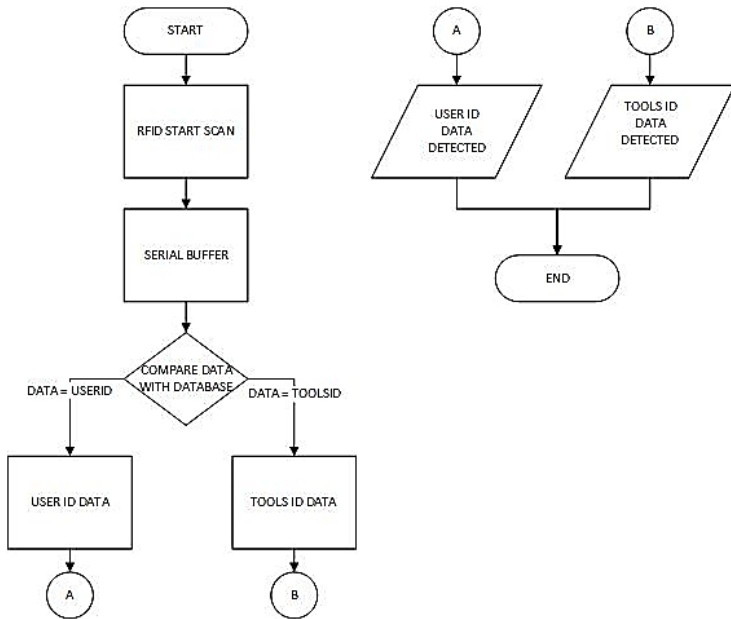


Gambar 3. 12 Diagram Alur Program Pengaturan Jarak Baca

3.4.2 Perancangan Program Pemilahan Tipe Data RFID

Data yang diterima oleh RFID pada saat scanning tidak mampu memilih tag yang akan diterima, sehingga memerlukan proses pemilihan data pada saat hasil pemindaian muncul. Kedua jenis tag yang digunakan yaitu *tag sticker* dan *tag kartu*, frekuensi yang digunakan untuk menerima data dalam *tag* termasuk dalam satu rentang frekuensi, sehingga kedua

jenis data terdeteksi dalam satu kali pemindaian. Data yang telah dipindai



Gambar 3. 13 Flowchart Pemilahan Data RFID

oleh RFID tercatat dalam *buffer* sehingga dapat diketahui data apa saja yang telah terdeteksi oleh interrogator. Data *Buffer* yang telah diolah dibandingkan dengan data yang ada dalam database sebagai acuan, sehingga dapat membedakan tipe *tag* apa yang terdeteksi oleh *reader*. Perancangan program pemilahan diaplikasikan pada tahap pemindaian *Tag* peralatan dan *Tag* pengguna, sehingga dalam proses pemindaian tidak terjadi kesalahan baca tipe data yang telah dicatat dalam buffer.

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISIS

Pembuatan alat ini perlu diketahui ketercapaian dari tujuan pembuatan, sehingga perlu dilakukan pengujian dan analisis terhadap alat yang telah dibuat. Pengujian terdiri dari pengukuran jarak baca dari RFID, pengujian komunikasi RFID, dan pengujian pengaturan otomatis pada RFID. Setelah dilakukan pengujian, dilakukan analisa terhadap bagian bagian alat yang telah diuji.

4.1 Pengujian Komunikasi RFID

Pengujian Komunikasi bertujuan untuk mengetahui kemampuan RFID untuk mengirimkan data serial dengan membaca tag RFID tipe EPCC1-G2. Pengujian dilakukan dengan membaca *tag* yang ada pada peralatan yang ada di *Automation Tool's Crib* dengan mengubah parameter serial, kekuatan pancaran, kapasitas baca, dan rentang frekuensi yang digunakan.

Tabel 4.1 Pengujian Komunikasi RFID

Baud Rate (bps)	Kekuatan (dB)	Frekuensi (MHz)	Jumlah alat tersedia	Alat yang terbaca
9600	5	902.6 - 927.4	7	0
	10			7
	20			7
	30			7
	5	920 – 924		0
	10			7
	20			7
	30			7
	5	865-867		0
	10			0
	20			0
	30			0
19200	5	902.6 - 927.4	7	0
	10			6
	20			7

Baud Rate (bps)	Kekuatan (dB)	Frekuensi (MHz)	Jumlah alat tersedia	Alat yang terbaca
	30	920 – 924		7
	5			0
	10			7
	20			7
	30			7
	5	865-867		0
	10			0
	20			0
	30			0
57600	5	902.6 - 927.4	7	1
	10			7
	20			7
	30			7
	5	920 – 924		0
	10			7
	20			7
	30			7
	5	865-867		0
	10			0
	20			0
	30			0

Dari tabel diatas merupakan data percobaan komunikasi data dari RFID dengan membaca tag yang telah ditempelkan di peralatan parameter yang telah diubah yang bervariasi digunakan untuk menentukan pengaturan komunikasi mana yang paling tepat digunakan. Dengan merubah variabel *baudrate* pada koneksi serial akan mempengaruhi kecepatan pengiriman data karena semakin tinggi *baudrate* maka akan semakin banyak modulasi yang dikirimkan. Pengaturan variable frekuensi digunakan untuk menentukan rentang frekuensi yang lebih tepat untuk digunakan pada tag. Pemilihan rentang frekuensi berdasarkan standar yang ada di Eropa, Amerika Serikat, dan Republik China. Variabel kontrol lain yang digunakan untuk mengukur komunikasi adalah *power* yang digunakan untuk mengatur daya pancar yang mempengaruhi jarak baca dari RFID.

Pemilihan metode komunikasi data pada RFID bertujuan untuk membandingkan konfigurasi yang ada pada komunikasi serial yang merupakan salah satu media yang dapat digunakan pada RFID. Penentuan konfigurasi pada RFID berujung pada preset dari kabel serial dengan kebutuhan penyesuaian kabel pada mesin. Perbedaan *baudrate* pada data yang diambil menentukan kecepatan pengiriman data serial dengan satuan bps (*bit per second*) yang semakin cepat seiring dengan semakin besar nilai *baudrate*. Pemilihan *baudrate* yang tepat mempengaruhi hasil data yang terbaca oleh RFID. Kecepatan pengiriman data akan mempengaruhi data yang terbaca oleh perangkat sehingga terjadi fenomena *missing bit* yang merubah data secara keseluruhan.

4.2 Pengujian Radiasi Jarak Baca RFID

Pengujian radiasi pada RFID dilakukan diruangan yang luas tanpa ada halangan di sekitar ruangan, dengan mengukur pola radiasi antenna berdasarkan daya pancar RFID di setiap 10° busur. Pengukuran tersebut bertujuan untuk menentukan konfigurasi daya pancar dan penempatan dari alat sehingga tepat untuk digunakan. Berikut adalah ilustrasi metode pengujian pada gambar 4.1 dan data hasil pengujian pada tabel 4.2. pengukuran dari setiap data sudut putar berupa data linier sehingga untuk menghitung intensitas radial menggunakan persamaan konversi daya linier ke radial. Direktivitas atau arah radiasi dari RFID dapat ditentukan menggunakan metode *radiation pattern* karena dengan menguji sensitivitas dari perangkat ditemukan wilayah dimana ada daya yang lebih besar di sisi depan RFID dibandingkan sisi belakang dari RFID. Metode ini juga dilakukan untuk menentukan daya yang sesuai untuk pengiriman dan pemindaian data

$$I_R = 10^{\frac{I_{lin}}{10}} \quad (4.1)$$

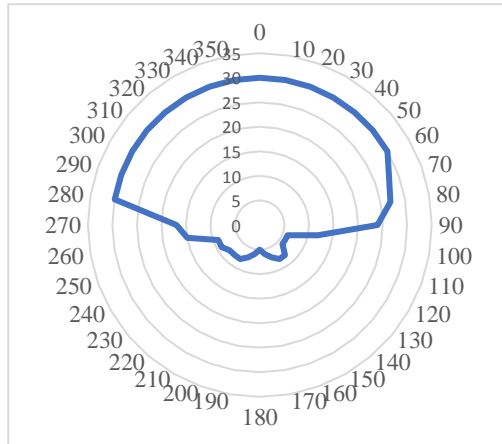
Tabel 4.2 Data Intensitas Daya

Sudut Putar (°)	Intensitas Linear (dB)	Intensitas Radial (dB)
0	30	1000
10	30	1000
20	30	1000
30	30	1000
40	30	1000

Sudut Putar (°)	Intensitas Linear (dB)	Intenstas Radial (dB)
50	30	1000
60	30	1000
70	28	630.957
80	27	501.187
90	24	251.189
100	12	15.8489
110	6	3.98107
120	6	3.98107
130	6	3.98107
140	8	6.30957
150	8	6.30957
160	7	5.01187
170	6	3.98107
180	5	3.16228
190	6	3.98107
200	7	5.01187
210	8	6.30957
220	8	6.30957
230	8	6.30957
240	9	7.94328
250	9	7.94328
260	15	31.6228
270	17	50.1187
280	30	1000
290	30	1000
300	30	1000
310	30	1000
320	30	1000
330	30	1000
340	30	1000
350	30	1000

Data yang telah diambil di setiap sudut untuk mengukur pola radiasi dari RFID, digambarkan menggunakan grafik pada Gambar 4.3 untuk mengetahui direktivitas dari RFID. Mengacu dari data yang telah diambil pada sudut yang merupakan sisi belakang dari RFID masih terdapat kemampuan dari RFID untuk mendeteksi *tag*, namun sangat kecil sekali kemungkinannya untuk digunakan secara sepenuhnya. *Side-lobe* dari RFID merupakan daya yang tidak diinginkan dari daya keseluruhan yang dikeluarkan oleh RFID. Direktivitas dari RFID yang telah ditentukan dari wilayah yang paling jauh dijangkau oleh RFID atau biasa disebut dengan *main lobe*. Memastikan *main-lobe* berada sesuai

dengan tujuan *tag* dan menyesuaikan agar wilayah yang dibaca tidak keluar dari area *tag* yang dituju. Oleh karena itu, data yang diperoleh untuk penentuan pola radiasi digambarkan dalam bentuk diagram melingkar untuk menggambarkan nilai



Gambar 4. 1 Pola Radiasi RFID UHF CT-I809

4.3 Pengujian Pemindaian Peralatan Kerja

Pemindaian peralatan dalam lemari untuk mengukur ketepatan pemindaian dari RFID, dengan meletakkan peralatan di posisi yang telah ditentukan dan pengujian ini menggunakan variable control berupa posisi rak yang ada dalam lemari untuk mengubah jarak reader dari peralatan. *Antenna Power* pada RFID adalah parameter untuk menentukan jangkauan maksimum dari alat. Lalu adanya *inventory scantime* digunakan untuk membatasi jumlah peralatan yang dibaca oleh RFID agar pemindaian lebih terfokus dan parameter *Read Interval* Berikut adalah tabel percobaan dengan posisi peralatan di rak paling atas. Perbedaan interval yang ditetapkan pada pengujian dilakukan untuk memberikan pengaruh dalam identifikasi peralatan yang ada dalam lemari. Variable tetap yang berupa daya antenna dan *inventory scantime* merupakan parameter untuk yang menjadi konfigurasi utama pengujian sehingga variabel telah didapatkan menggunakan data dari pengujian sebelumnya untuk menyesuaikan data yang ada dalam lemari.

Pengujian ini diharapkan dapat memberikan hasil yang dapat mengurangi peluang kesalahan yang terjadi pada saat fase identifikasi. Kesalahan yang terjadi sangat vital apabila terjadi pada proses identifikasi karena mempengaruhi data yang telah tercatat dalam database.

Tabel 4.3 Pemindaian dengan Interval Waktu 50ms

No	Antenna Power (dB)	Inventory Scantime (detik)	Read interval (ms)	Peralatan dalam lemari	Peralatan yang terdeteksi
1	15	2	50	12	10
2	15	2	50	12	12
3	15	2	50	12	12
4	15	2	50	12	12
5	15	2	50	12	10
6	15	2	50	12	12
7	15	2	50	12	12
8	15	2	50	12	12
9	15	2	50	12	12
10	15	2	50	12	12

Dari tabel 4.3, jumlah peralatan yang terdeteksi oleh RFID terkadang mengalami kegagalan pemindaian sehingga terdapat peralatan yang tidak terdeteksi.

$$\begin{aligned}
 \text{Error pemindaian} &= \frac{\text{Rata Rata Alat terdeteksi}}{\text{Jumlah Alat terdeteksi}} \times 100 \% = \frac{11.1}{12} \times 100 \% \\
 &= 92.5\%
 \end{aligned}$$

Data berikut ini merupakan data pemindaian di posisi peralatan yang sama dengan mempercepat waktu interval dari pembacaan dari 50 ms ke 30 ms.

Tabel 4.4 Pemindaian Peralatan dengan Interval Waktu 30ms.

No	Antenna Power (dB)	Inventory Scantime (detik)	Read interval (ms)	Peralatan dalam lemari	Peralatan yang terdeteksi
1	15	2	30	12	12

No	Antenna Power (dB)	Inventory Scantime (detik)	Read interval (ms)	Peralatan dalam lemari	Peralatan yang terdeteksi
2	15	2	30	12	12
3	15	2	30	12	12
4	15	2	30	12	12
5	15	2	30	12	12
6	15	2	30	12	12
7	15	2	30	12	11
8	15	2	30	12	12
9	15	2	30	12	10
10	15	2	30	12	12

$$\begin{aligned}
 \text{Error pemindaian} &= \frac{\text{Rata Rata Alat terdeteksi}}{\text{Jumlah Alat terdeteksi}} \times 100\% = \frac{11.8}{12} \times 100\% \\
 &= 98.3\%
 \end{aligned}$$

Perubahan jumlah peralatan pada Tabel 4.4 yang terdeteksi dalam lemari masih terdapat kegagalan pemindaian dari RFID untuk memindai data peralatan secara utuh. *Tag* yang terpasang pada peralatan berpengaruh pada pemindaian *reader* dari data yang diperoleh dengan metode uji pindai. Data yang didapatkan dari pengujian menghasilkan data yang berpengaruh pada penempatan peralatan dibawah RFID.

4.4 Pengujian Prosedur Sistem *Automation Tool's Crib*

Sistem transaksi dan identifikasi yang telah dirancang menyesuaikan dengan kebiasaan dan efektivitas dari pengguna, prosedur yang telah kami buat perlu diuji ketahanan dari penggunaan. Uji ketahanan biasa dilakukan untuk menemukan masalah yang ada jika terdapat komponen atau komunikasi yang berpengaruh akibat adanya ketahanan yang kurang pada saat mesin dioperasikan. Pengujian prosedur dilakukan dengan melakukan transaksi sekaligus identifikasi pada proses peminjaman dan pengembalian peralatan pada sistem. Pengguna yang hendak meminjam peralatan yang ada dalam lemari terlihat masih kaku menggunakan kartu *tag* yang masih awam bagi pengguna. Mendekatkan

kartu sesuai dengan jarak baca dari RFID membuat akses pada mesin dapat langsung menampilkan data pengguna serta riwayat peralatan yang telah dipinjam oleh pengguna. Setelah proses identifikasi selesai dilakukan proses transaksi dengan mengambil atau mengembalikan peralatan yang ada pada lemari. Tahap transaksi menimbulkan pertanyaan serta pengecualian bagi pengguna dikarenakan posisi pengguna yang terlalu dekat dengan alat pada saat pemindaian ulang oleh RFID dan pengembalian alat tidak sesuai pada tempatnya juga akan menimbulkan kesalahan baca pada peralatan.

4.5 Buku Manual *Automation Tool's Crib*

Pembuatan buku panduan dari sistem merupakan persyaratan dari pembuatan mesin di PT. Astra Otoparts Divisi WINTEQ. Buku panduan dari sistem terdiri dari prosedur keamanan untuk menggunakan sistem. lalu dilanjutkan dengan deskripsi mesin yang meliputi dimensi, spesifikasi dan fitur yang ada dalam mesin. Prosedur operasional dari mesin juga merupakan konten dari buku manual dikarenakan digunakan untuk memberikan pengertian kepada pengguna dan administrator untuk mengakses data yang ada dalam PC. Bab kedua dari buku manual merupakan cara penggunaan dari sistem bagi pengguna maupun *administrator*. Cara penggunaan dari sistem memiliki prosedur khusus yang menunjang kelancaran dari alat untuk memperlakukan sistem yang otomatis. Contohnya, jarak terdekat dan terjauh dari pemindaian pengguna bagi karyawan yang ingin mengakses sistem memerlukan jarak terjauh hanya sampai 45 cm dari lemari karena merupakan jarak paling ideal bagi RFID untuk membaca kartu akses. Bab terakhir dari buku manual dari sistem ini adalah proses *troubleshooting* pada sistem jika terjadi permasalahan yang menyangkut proses transaksi dan identifikasi dari peralatan.

Tabel4. 5 Komponen Automation Tool's Crib

No	Deskripsi komponen
1	Kunci Pintu Solenoid Door Lock 12 V Automatic Electronic Arduino
2	KR18268 Long Lever Limit Switch 15 A 250VAC//#24
3	RFID UHF Windshield Sticker Passive Tag EPC Gen2/ISO18000-6C
4	UHF RFID Passive on-metal Tag Screw holes 915MHZ

No	Deskripsi komponen
5	Kartu RFID Card UHF ISO18000-6C
6	UHF Reader & Writer (Middle Range)-CT-I809
7	Selector Key Switch 3 Posisi Stay Put
8	Power Supply 12V 3A Jaring CCTV LED AC 220V to DC 12V 3A Adaptor PSU
9	DR22D0L-E3G_FUJI_PILOT LAMP GREEN
10	DR22D0L-E3R_FUJI_PILOT LAMP RED
11	DR22D0L-E3Y_FUJI_PILOT LAMP YELLOW
12	MORGAN BOOKCASE_INFORMA_WHITE BOOKCASE (216.2X43.4X186)
13	UTICON_POWER Outlet Isi 6
14	Kabel Serial DB9 male-female 10M
15	UC-232_ATEN_USB to Serial Converter
16	Kabel LAN UTP RJ45 10 M
17	BOX MCB 2 POLE(BW32AAG)
18	Legrand rubber 2P+E POWER PLUG
19	DURA 600_OMNI-ID RFID TEXT
20	TL5 T5_PHILIPS_LAMPU LED 4W
21	CJ1M - OMRON - PROGRAMMABLE LOGIC CONTROLLER

Sehingga, kepemilikan buku manual pada sistem akan memberikan wawasan secara umum dan mendalam dari mesin yang akan digunakan. Pada Tabel 4.5 merupakan komponen dari sistem yang digunakan untuk membangun sistem secara keseluruhan. Komponen tersebut merupakan hasil dari persyaratan sistem yang harus dilampaui dan dicapai sebagai titik keberhasilan penyelesaian masalah. Penentuan jenis dan jumlah komponen dilakukan untuk memenuhi *design requirement* yang telah direncanakan pada sistem. Namun, Dilakukan serah terima buku manual serta pelatihan kepada administrator untuk menyerahkan sistem sepenuhnya sehingga dapat digunakan untuk produksi dan melakukan riset.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Berdasarkan hasil pengujian dan analisa yang telah dilakukan dapat diperoleh kesimpulan bahwa penggunaan RFID sebagai sistem identifikasi pengguna dan peralatan pada *Automation tool's crib* meningkatkan efektivitas dan menghemat waktu proses. Proses pemilahan data menggunakan *database* Microsoft Access sebagai database berhasil memisahkan data pengguna dan peralatan yang setiap saat terdeteksi oleh *reader* untuk dipisahkan sesuai tahap prosedur. Hasil pengukuran pola radiasi dari UHF RFID membantu mempermudah pengaturan jarak baca dari alat yang sebelumnya prosentase *error* sebesar 7,5% menjadi 1,6% mengurangi terjadinya kesalahan baca pada alat. Penelitian selanjutnya dari sistem disarankan menggunakan *tag* yang khusus pada peralatan kerja serta RFID tipe UHF yang mampu beradaptasi dengan kondisi lingkungan di industri yang penuh interferensi agar sistem *Automation Tool's Crib* dapat digunakan sepenuhnya menggantikan *Tool's Crib* konvensional

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] E. Schaupp, E. Abele, and J. Metternich, "Evaluating Relevant Factors for Developing an Optimal Tool Storage Strategy," *Procedia CIRP*, vol. 55, pp. 23–28, 2016.
- [2] J. Tanchoco, *Material flow systems in manufacturing*. 1994.
- [3] Badan Pusat Statistik, "Analisis Hasil Listing Sensus Ekonomi 2016 - Aglomerasi Industri Manufaktur di Indonesia," Jakarta, 2017.
- [4] C. Saygin, "Adaptive inventory management using RFID data," *Int. J. Adv. Manuf. Technol.*, vol. 32, no. 9–10, pp. 1045–1051, 2007.
- [5] S. Graham T., *Cutting Tool Technology*, no. 1. 2008.
- [6] D. Dobkin, *The RF in RFID*. 2008.
- [7] X. Zhao, Y. Huang, J. Li, Q. Zhang, and G. Wen, "Wideband high gain circularly polarized UHF RFID reader microstrip antenna and array," *AEU - Int. J. Electron. Commun.*, vol. 77, pp. 76–81, 2017.
- [8] A. Ramos, A. Lazaro, D. Girbau, and R. Villarino, "Introduction to RFID and Chipless RFID BT - RFID and Wireless Sensors Using Ultra-Wideband Technology," 2016, pp. 1–18.
- [9] A. Ramos, A. Lazaro, D. Girbau, and R. Villarino, *RFID and Wireless Sensors Using Ultra-Wideband Technology*. 2016.
- [10] Z. Qi, Y. Zhuang, X. Li, W. Liu, Y. Du, and B. Wang, "Full passive UHF RFID Tag with an ultra-low power, small area, high resolution temperature sensor suitable for environment monitoring," *Microelectronics J.*, vol. 45, no. 1, pp. 126–131, 2014.
- [11] M. Omer and G. Y. Tian, "Indoor distance estimation for passive UHF RFID tag based on RSSI and RCS," *Measurement*, vol. 127, no. May, pp. 425–430, 2018.
- [12] J. S. Sun and C. H. Wu, "A broadband circularly polarized

- antenna of square-ring patch for UHF RFID reader applications,” *AEU - Int. J. Electron. Commun.*, vol. 85, no. December 2017, pp. 84–90, 2018.
- [13] A. Aguilar-Garcia, S. Fortes, E. Colin, and R. Barco, “Enhancing RFID indoor localization with cellular technologies,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2015, no. 1, 2015.
- [14] C. H. Hsu, Y. M. Chen, and H. J. Kang, “Performance-effective and low-complexity redundant reader detection in wireless RFID networks,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2008, 2008.

LAMPIRAN

Lampiran 1. Program RFID

`Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load`

```
list = New ArrayList()  
progressBar1.Visible = False  
fOpenComIndex = -1  
fComAdr = 0  
ferrorcode = -1  
fBaud = 5  
InitComList()  
InitReaderList()  
NoAlarm_G2.Checked = True
```

```
Byone_6B.Checked = True  
Different_6B.Checked = True
```

```
P_EPC.Checked = True  
C_EPC.Checked = True  
DestroyCode.Checked = True  
NoProect.Checked = True  
NoProect2.Checked = True  
fAppClosed = False  
flsInventoryScan = False  
fisinventoryscan_6B = False  
fTimer_6B_ReadWrite = False  
Label_Alarm.Visible = False  
Timer_Test_.Enabled = False  
Timer_G2_Read.Enabled = False  
Timer_G2_Alarm.Enabled = False  
timer1.Enabled = False
```

```
Button3.Enabled = False  
button20.Enabled = False  
Button5.Enabled = False  
Button1.Enabled = False  
button2.Enabled = False
```

Button_DestroyCard.Enabled = False
Button_WriteEPC_G2.Enabled = False
Button_SetReadProtect_G2.Enabled = False
Button_SetMultiReadProtect_G2.Enabled = False
Button_RemoveReadProtect_G2.Enabled = False
Button_CheckReadProtected_G2.Enabled = False
Button_SetEASAlarm_G2.Enabled = False
button4.Enabled = False
Button_LockUserBlock_G2.Enabled = False
SpeedButton_Read_G2.Enabled = False
Button_DataWrite.Enabled = False
Button_BlockErase.Enabled = False
Button_BlockWrite.Enabled = False
Button_SetProtectState.Enabled = False
SpeedButton_Query_6B.Enabled = False
SpeedButton_Read_6B.Enabled = False
SpeedButton_Write_6B.Enabled = False
Button14.Enabled = False
Button15.Enabled = False

DestroyCode.Enabled = False
AccessCode.Enabled = False
NoProect.Enabled = False
Proect.Enabled = False
Always.Enabled = False
AlwaysNot.Enabled = False
NoProect2.Enabled = False
Proect2.Enabled = False
Always2.Enabled = False
AlwaysNot2.Enabled = False
P_Reserve.Enabled = False
P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Same_6B.Enabled = False
Different_6B.Enabled = False
Less_6B.Enabled = False
Greater_6B.Enabled = False


```

radioButton1.Checked = True
radioButton4.Checked = True
radioButton5.Checked = True
radioButton7.Checked = True
radioButton10.Checked = True
radioButton14.Checked = True
button6.Enabled = False
button8.Enabled = False
button9.Enabled = False
button10.Enabled = False
button11.Enabled = False
comboBox5.Enabled = False
radioButton5.Enabled = False
radioButton6.Enabled = False
radioButton7.Enabled = False
radioButton8.Enabled = False
radioButton9.Enabled = False
radioButton10.Enabled = False
radioButton11.Enabled = False
radioButton12.Enabled = False
radioButton13.Enabled = False
radioButton14.Enabled = False
radioButton15.Enabled = False
RadioButton16.Enabled = False
RadioButton17.Enabled = False
RadioButton19.Enabled = False
textBox3.Enabled = False
radioButton_band1.Checked = True
RadioButton16.Checked = True
ComboBox_baud2.SelectedIndex = 3
comboBox9.SelectedIndex = 0
comboBox10.SelectedIndex = 0
radioButton22.Checked = True
If Me IsNot Nothing Then
    automatic()
End If
End Sub

```

```

Private Sub OpenPort_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OpenPort.Click
    Dim port As Integer = 0
    Dim openresult, i As Integer
    Dim temp As String
    openresult = 30
    Cursor = Cursors.WaitCursor
    If (Edit_CmdComAddr.Text = "") Then
        Edit_CmdComAddr.Text = "FF"
    End If
    fComAdr = Convert.ToByte(Edit_CmdComAddr.Text, 16)
    Try
        If (ComboBox_COM.SelectedIndex = 0) Then
            fBaud =
Convert.ToByte(ComboBox_baud2.SelectedIndex)
            If (fBaud > 2) Then
                fBaud = Convert.ToByte(fBaud + 2)
            End If
            openresult =
StaticClassReaderB.AutoOpenComPort(port, fComAdr, fBaud,
frmcomportindex)
            fOpenComIndex = frmcomportindex
            If (openresult = 0) Then
                ComOpen = True
                If (fBaud > 3) Then
                    ComboBox_baud.SelectedIndex =
Convert.ToInt32(fBaud - 2)
                Else
                    ComboBox_baud.SelectedIndex =
Convert.ToInt32(fBaud)
                End If
                Button3_Click(sender, e)
                If ((openresult = &H35) Or (openresult = &H30)) Then
                    MessageBox.Show("Serial Communication Error or
Occupied", "Information")
                End If
                StaticClassReaderB.CloseSpecComPort(frmcomportindex)
                ComOpen = False
            End If
        End If
    End Try

```

```

        End If
    Else
        temp = ComboBox_COM.SelectedItem.ToString()
        temp = temp.Trim()
        port = Convert.ToInt32(temp.Substring(3, temp.Length -
3))

        For i = 6 To 1 Step -1
            fBaud = Convert.ToByte(i)
            If (fBaud = 3) Then
                Continue For
            End If
            openresult = StaticClassReaderB.OpenComPort(port,
fComAdr, fBaud, frmcomportindex)
            fOpenComIndex = frmcomportindex
            If (openresult = &H35) Then
                MessageBox.Show("COM Opened", "Information")
                Exit Sub
            End If
            If (openresult = 0) Then
                ComOpen = True
                Button3_Click(sender, e)
                If (fBaud > 3) Then
                    ComboBox_baud.SelectedIndex =
Convert.ToInt32(fBaud - 2)
                Else
                    ComboBox_baud.SelectedIndex =
Convert.ToInt32(fBaud)
                End If
                If ((openresult = &H35) Or (openresult = &H30))
Then
                    ComOpen = False
                    MessageBox.Show("Serial Communication Error
or Occupied", "Information")

                StaticClassReaderB.CloseSpecComPort(frmcomportindex)
                Exit Sub
            End If
            RefreshStatus()
        Exit For
    
```

```

        End If
    Next i

    End If
Finally
    Cursor = Cursors.Default
End Try
If ((fOpenComIndex <> -1) And (openresult <> &H35) And
(openresult <> &H30)) Then
    ComboBox_AlreadyOpenCOM.Items.Add("COM"      +
Convert.ToString(fOpenComIndex))
    ComboBox_AlreadyOpenCOM.SelectedIndex      =
ComboBox_AlreadyOpenCOM.SelectedIndex + 1
    Button3.Enabled = True
    button20.Enabled = True
    Button5.Enabled = True
    Button1.Enabled = True
    button2.Enabled = True
    Button_WriteEPC_G2.Enabled = True
    Button_SetReadProtect_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True
    button4.Enabled = True
    SpeedButton_Query_6B.Enabled = True
    button6.Enabled = True
    button8.Enabled = True
    button9.Enabled = True
    Button12.Enabled = True
    button_OffsetTime.Enabled = True
    button_settigtime.Enabled = True
    button_gettigtime.Enabled = True
    ComOpen = True
    button10.Text = "Get"
End If
If ((fOpenComIndex = -1) And (openresult = &H30)) Then
    MessageBox.Show("Serial Communication Error or
Occupied", "Information")
End If

```

```

        If ((ComboBox_AlreadyOpenCOM.Items.Count <> 0) And
(fOpenComIndex <> -1) And (openresult <> &H35) And (openresult <>
&H30) And (openresult = 0)) Then

```

```

        fComAdr = Convert.ToByte(Edit_ComAdr.Text, 16)
        temp =
ComboBox_AlreadyOpenCOM.SelectedItem.ToString()
        frmcomportindex = Convert.ToInt32(temp.Substring(3,
temp.Length - 3))
        End If
        RefreshStatus()
    End Sub

```

```

    Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click

```

```

        Dim TrType(2) As Byte
        Dim VersionInfo(2) As Byte
        Dim ReaderType As Byte = 0
        Dim ScanTime As Byte = 0
        Dim dmaxfre As Byte = 0
        Dim dminfre As Byte = 0
        Dim powerdBm As Byte = 0
        Dim FreBand As Byte = 0
        Dim i As Integer = 0
        Edit_Version.Text = ""
        Edit_ComAdr.Text = ""
        Edit_scantime.Text = ""
        Edit_Type.Text = ""
        ISO180006B.Checked = False
        EPCC1G2.Checked = False
        Edit_powerdBm.Text = ""
        Edit_dminfre.Text = ""
        Edit_dmaxfre.Text = ""
        ComboBox_PowerDbm.Items.Clear()
        fCmdRet =
StaticClassReaderB.GetReaderInformation(fComAdr, VersionInfo,
ReaderType, TrType, dmaxfre, dminfre, powerdBm, ScanTime,
frmcomportindex)
        If (fCmdRet = 0) Then

```

```

        Edit_Version.Text = Convert.ToString(VersionInfo(0),
10).PadLeft(2, "0") + "." + Convert.ToString(VersionInfo(1),
10).PadLeft(2, "0")
        If (VersionInfo(1) >= 30) Then
            For i = 0 To 30

ComboBox_PowerDbm.Items.Add(Convert.ToString(i))
            Next i
            If (powerdBm > 30) Then
                ComboBox_PowerDbm.SelectedIndex = 30
            Else
                ComboBox_PowerDbm.SelectedIndex = powerdBm
            End If
        Else
            For i = 0 To 18

ComboBox_PowerDbm.Items.Add(Convert.ToString(i))
            Next i
            If (powerdBm > 18) Then
                ComboBox_PowerDbm.SelectedIndex = 18
            Else
                ComboBox_PowerDbm.SelectedIndex = powerdBm
            End If
        End If
        Edit_ComAdr.Text = Convert.ToString(fComAdr,
16).PadLeft(2, "0")
        Edit_NewComAdr.Text = Convert.ToString(fComAdr,
16).PadLeft(2, "0")
        Edit_scantime.Text = Convert.ToString(ScanTime,
10).PadLeft(2, "0") + "*100ms"
        ComboBox_scantime.SelectedIndex = ScanTime - 3
        Edit_powerdBm.Text = Convert.ToString(powerdBm,
10).PadLeft(2, "0")
        FreBand = Convert.ToByte((((dmaxfre And &HC0) >> 4) Or
(dminfre >> 6))
        Select Case FreBand
            Case 0
                radioButton_band1.Checked = True

```

```

    fdminfre = 902.6 + (dminfre And &H3F) * 0.4
    fdmaxfre = 902.6 + (dmaxfre And &H3F) * 0.4
Case 1
    radioButton_band2.Checked = True
    fdminfre = 920.125 + (dminfre And &H3F) * 0.25
    fdmaxfre = 920.125 + (dmaxfre And &H3F) * 0.25
Case 2
    radioButton_band3.Checked = True
    fdminfre = 902.75 + (dminfre And &H3F) * 0.5
    fdmaxfre = 902.75 + (dmaxfre And &H3F) * 0.5
Case 3
    radioButton_band4.Checked = True
    fdminfre = 917.1 + (dminfre And &H3F) * 0.2
    fdmaxfre = 917.1 + (dmaxfre And &H3F) * 0.2
Case 4
    radioButton_band5.Checked = True
    fdminfre = 865.1 + (dminfre And &H3F) * 0.2
    fdmaxfre = 865.1 + (dmaxfre And &H3F) * 0.2
Case Else

End Select
Edit_dminfre.Text = Convert.ToString(fdminfre) + "MHz"
Edit_dmaxfre.Text = Convert.ToString(fdmaxfre) + "MHz"
If (fdmaxfre <> fdminfre) Then
    CheckBox_SameFre.Checked = False
End If
ComboBox_dminfre.SelectedIndex = dminfre And &H3F
ComboBox_dmaxfre.SelectedIndex = dmaxfre And &H3F
If (ReaderType = &H3) Then
    Edit_Type.Text = ""
End If
If (ReaderType = &H6) Then
    Edit_Type.Text = ""
End If
If (ReaderType = &H9) Then
    Edit_Type.Text = "UHFReader18"
End If
If ((TrType(0) And &H2) = &H2) Then
    ISO180006B.Checked = True

```

```

        EPCC1G2.Checked = True
    Else
        ISO180006B.Checked = False
        EPCC1G2.Checked = False
    End If
End If
AddCmdLog("GetReaderInformation",
"GetReaderInformation", fCmdRet)
End Sub

Private Sub ClosePort_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ClosePort.Click
    Dim port As Integer
    'string SelectCom ;
    Dim temp As String
    ClearLastInfo()
    Try
        If (ComboBox_AlreadyOpenCOM.SelectedIndex < 0) Then
            MessageBox.Show("Please Choose COM Port to close",
"Information")
        Else
            temp =
ComboBox_AlreadyOpenCOM.SelectedItem.ToString()
            port = Convert.ToInt32(temp.Substring(3, temp.Length -
3))
            fCmdRet = StaticClassReaderB.CloseSpecComPort(port)
            If (fCmdRet = 0) Then
                ComboBox_AlreadyOpenCOM.Items.RemoveAt(0)
                If (ComboBox_AlreadyOpenCOM.Items.Count <> 0)
Then
                    temp =
ComboBox_AlreadyOpenCOM.SelectedItem.ToString()
                    port = Convert.ToInt32(temp.Substring(3,
temp.Length - 3))
                    StaticClassReaderB.CloseSpecComPort(port)
                    fComAdr = &HFF
                    StaticClassReaderB.OpenComPort(port, fComAdr,
fBaud, frmcomportindex)
                    fOpenComIndex = frmcomportindex
                End If
            End If
        End If
    Catch ex As Exception
        AddCmdLog("Error", ex.Message)
    End Try
End Sub

```



```

RefreshStatus()
Button3_Click(sender,
'×Ô¶- ÖÐ¶ÁÈ;Ð';;Æ÷ÐÃ¶
End If
Else
    MessageBox.Show("Serial Communication Error",
"Information")
End If
End If
Finally

End Try
If (ComboBox_AlreadyOpenCOM.Items.Count <> 0) Then
    ComboBox_AlreadyOpenCOM.SelectedIndex = 0
Else
    fOpenComIndex = -1
    ComboBox_AlreadyOpenCOM.Items.Clear()
    ComboBox_AlreadyOpenCOM.Refresh()
    RefreshStatus()
    Button3.Enabled = False
    button20.Enabled = False
    Button5.Enabled = False
    Button1.Enabled = False
    button2.Enabled = False
    Button_DestroyCard.Enabled = False
    Button_WriteEPC_G2.Enabled = False
    Button_SetReadProtect_G2.Enabled = False
    Button_SetMultiReadProtect_G2.Enabled = False
    Button_RemoveReadProtect_G2.Enabled = False
    Button_CheckReadProtected_G2.Enabled = False
    Button_SetEASAlarm_G2.Enabled = False
    button4.Enabled = False
    Button_LockUserBlock_G2.Enabled = False
    SpeedButton_Read_G2.Enabled = False
    Button_DataWrite.Enabled = False
    Button_BlockErase.Enabled = False
    Button_BlockWrite.Enabled = False
    Button_SetProtectState.Enabled = False
    SpeedButton_Query_6B.Enabled = False

```

SpeedButton_Read_6B.Enabled = False
SpeedButton_Write_6B.Enabled = False
Button14.Enabled = False
Button15.Enabled = False

DestroyCode.Enabled = False
AccessCode.Enabled = False
NoProect.Enabled = False
Proect.Enabled = False
Always.Enabled = False
AlwaysNot.Enabled = False
NoProect2.Enabled = False
Proect2.Enabled = False
Always2.Enabled = False
AlwaysNot2.Enabled = False

P_Reserve.Enabled = False
P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Alarm_G2.Enabled = False
NoAlarm_G2.Enabled = False

Same_6B.Enabled = False
Different_6B.Enabled = False
Less_6B.Enabled = False
Greater_6B.Enabled = False
button6.Enabled = False
button8.Enabled = False
button9.Enabled = False

DestroyCode.Enabled = False
AccessCode.Enabled = False
NoProect.Enabled = False
Proect.Enabled = False
Always.Enabled = False
AlwaysNot.Enabled = False
NoProect2.Enabled = False
Proect2.Enabled = False

Always2.Enabled = False
AlwaysNot2.Enabled = False
P_Reserve.Enabled = False
P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Button_WriteEPC_G2.Enabled = False
Button_SetMultiReadProtect_G2.Enabled = False
Button_RemoveReadProtect_G2.Enabled = False
Button_CheckReadProtected_G2.Enabled = False
button4.Enabled = False

Button_DestroyCard.Enabled = False
Button_SetReadProtect_G2.Enabled = False
Button_SetEASAlarm_G2.Enabled = False
Alarm_G2.Enabled = False
NoAlarm_G2.Enabled = False
Button_LockUserBlock_G2.Enabled = False
SpeedButton_Read_G2.Enabled = False
Button_DataWrite.Enabled = False
Button_BlockWrite.Enabled = False
Button_BlockErase.Enabled = False
Button_BlockWrite.Enabled = False
Button_SetProtectState.Enabled = False
ListView1_EPC.Items.Clear()
ComboBox_EPC1.Items.Clear()
ComboBox_EPC2.Items.Clear()
ComboBox_EPC3.Items.Clear()
ComboBox_EPC4.Items.Clear()
ComboBox_EPC5.Items.Clear()
ComboBox_EPC6.Items.Clear()
checkBox1.Enabled = False
button10.Text = "Get"
SpeedButton_Read_6B.Enabled = False
SpeedButton_Write_6B.Enabled = False
Button14.Enabled = False
Button15.Enabled = False
ListView_ID_6B.Items.Clear()
ComOpen = False

```

        timer1.Enabled = False
        button10.Enabled = False
        comboBox4.SelectedIndex = 0
        Button12.Enabled = False
        button_OffsetTime.Enabled = False
        button_settigitime.Enabled = False
        button_gettigitime.Enabled = False
    End If
End Sub

Private Sub Button5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button5.Click
    Dim aNewComAdr, powerDbm, dminfre, dmaxfre, scantime,
band As Byte
    Dim returninfo As String = ""
    Dim returninfoDlg As String = ""
    Dim setinfo As String
    If (radioButton_band1.Checked) Then
        band = 0
    End If
    If (radioButton_band2.Checked) Then
        band = 1
    End If
    If (radioButton_band3.Checked) Then
        band = 2
    End If
    If (radioButton_band4.Checked) Then
        band = 3
    End If
    If (radioButton_band5.Checked) Then
        band = 4
    End If
    If (Edit_NewComAdr.Text = "") Then
        Exit Sub
    End If
    progressBar1.Visible = True
    progressBar1.Minimum = 0
    dminfre = Convert.ToByte(((band And 3) << 6) Or
(ComboBox_dminfre.SelectedIndex And &H3F))

```

```

        dmaxfre = Convert.ToByte(((band And &HC) << 4) Or
(ComboBox_dmaxfre.SelectedIndex And &H3F))
        aNewComAdr = Convert.ToByte(Edit_NewComAdr.Text)
        powerDbm =
Convert.ToByte(ComboBox_PowerDbm.SelectedIndex)
        fBaud = Convert.ToByte(ComboBox_baud.SelectedIndex)
        If (fBaud > 2) Then
            fBaud = Convert.ToByte(fBaud + 2)
        End If
        scantime =
Convert.ToByte(ComboBox_scantime.SelectedIndex + 3)
        setinfo = "Write"
        progressBar1.Value = 10
        fCmdRet = StaticClassReaderB.WriteComAdr(fComAdr,
aNewComAdr, frmcomportindex)
        If (fCmdRet = &H13) Then
            fComAdr = aNewComAdr
        End If
        If (fCmdRet = 0) Then
            fComAdr = aNewComAdr
            returninfo = returninfo + setinfo + "Address Successfully"
        ElseIf (fCmdRet = &HEE) Then
            returninfo = returninfo + setinfo + "Address Response
Command Error"
        Else
            returninfo = returninfo + setinfo + "Address Fail"
            returninfoDlg = returninfoDlg + setinfo + "Address Fail
Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
        End If
        progressBar1.Value = 25
        fCmdRet = StaticClassReaderB.SetPowerDbm(fComAdr,
powerDbm, frmcomportindex)
        If (fCmdRet = 0) Then
            returninfo = returninfo + ",Power Success"
        ElseIf (fCmdRet = &HEE) Then
            returninfo = returninfo + ",Power Response Command
Error"
        Else

```

```

        returninfo = returninfo + ",Power Fail"
        returninfoDlg = returninfoDlg + " " + setinfo + "Power Fail
Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
End If

```

```

        progressBar1.Value = 40
        fCmdRet = StaticClassReaderB.Writedfre(fComAdr, dmaxfre,
dminfre, frmcomportindex)
        If (fCmdRet = 0) Then
            returninfo = returninfo + ",Frequency Success"
        ElseIf (fCmdRet = &HEE) Then
            returninfo = returninfo + ",Frequency Response Command
Error"
        Else

```

```

            returninfo = returninfo + ",Frequency Fail"
            returninfoDlg = returninfoDlg + " " + setinfo + "Frequency
Fail Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
        End If

```

```

        progressBar1.Value = 55
        fCmdRet = StaticClassReaderB.Writebaud(fComAdr, fBaud,
frmcomportindex)
        If (fCmdRet = 0) Then
            returninfo = returninfo + ",Baud Rate Success"
        ElseIf (fCmdRet = &HEE) Then
            returninfo = returninfo + ",Baud Rate Response Command
Error"
        Else

```

```

            returninfo = returninfo + ",Baud Rate Fail"
            returninfoDlg = returninfoDlg + " " + setinfo + "Baud Rate
Fail Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
        End If

```

```

        progressBar1.Value = 70

```

```

        fCmdRet = StaticClassReaderB.WriteScanTime(fComAdr,
scantime, frmcomportindex)
    If (fCmdRet = 0) Then
        returninfo = returninfo + ",InventoryScanTime Success"
    ElseIf (fCmdRet = &HEE) Then
        returninfo = returninfo + ",InventoryScanTime Response
Command Error"
    Else
        returninfo = returninfo + ",InventoryScanTime Fail"
        returninfoDlg = returninfoDlg + " " + setinfo +
"InventoryScanTime Fail Command Response=0x" +
Convert.ToString(fCmdRet) + "(" + GetReturnCodeDesc(fCmdRet) + ")"
    End If

    progressBar1.Value = 100
    Button3_Click(sender, e)
    progressBar1.Visible = False
    StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + returninfo
    If (returninfoDlg <> "") Then
        MessageBox.Show(returninfoDlg, "Information")
    End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim aNewComAdr, powerDbm, dminfre, dmaxfre, scantime
As Byte
    Dim returninfo As String = ""
    Dim returninfoDlg As String = ""
    Dim setinfo As String
    progressBar1.Visible = True
    progressBar1.Minimum = 0
    dminfre = 0
    dmaxfre = 62
    aNewComAdr = &H0
    If (Convert.ToInt32(Edit_Version.Text.Substring(3, 2))) Then
        powerDbm = 30
    Else

```

```

        powerDbm = 18
    End If

    fBaud = 5
    scantime = 10
    setinfo = " Recovery "
    ComboBox_baud.SelectedIndex = 3
    progressBar1.Value = 10
    fCmdRet = StaticClassReaderB.WriteComAdr(fComAdr,
aNewComAdr, frmcomportindex)
    If (fCmdRet = &H13) Then
        fComAdr = aNewComAdr
    End If
    If (fCmdRet = 0) Then

        fComAdr = aNewComAdr
        returninfo = returninfo + setinfo + "Address Successfully"
    ElseIf (fCmdRet = &HEE) Then
        returninfo = returninfo + setinfo + "Address Response
Command Error"
    Else

        returninfo = returninfo + setinfo + "Address Fail"
        returninfoDlg = returninfoDlg + setinfo + "Address Fail
Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
    End If

    progressBar1.Value = 25
    fCmdRet = StaticClassReaderB.SetPowerDbm(fComAdr,
powerDbm, frmcomportindex)
    If (fCmdRet = 0) Then
        returninfo = returninfo + ",Power Success"
    ElseIf (fCmdRet = &HEE) Then
        returninfo = returninfo + ",Power Response Command
Error"
    Else
        returninfo = returninfo + ",Power Fail"

```



```

        returninfoDlg = returninfoDlg + " " + setinfo + "Power Fail
Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
    End If
    progressBar1.Value = 40
    fCmdRet = StaticClassReaderB.Writedfre(fComAdr, dmaxfre,
dminfre, frmcomportindex)
    If (fCmdRet = 0) Then
        returninfo = returninfo + ",Frequency Success"
    ElseIf (fCmdRet = &HEE) Then
        returninfo = returninfo + ",Frequency Response Command
Error"
    Else
        returninfo = returninfo + ",Frequency Fail"
        returninfoDlg = returninfoDlg + " " + setinfo + "Frequency
Fail Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
    End If

    progressBar1.Value = 55
    fCmdRet = StaticClassReaderB.Writebaud(fComAdr, fBaud,
frmcomportindex)
    If (fCmdRet = 0) Then
        returninfo = returninfo + ",Baud Rate Success"
    ElseIf (fCmdRet = &HEE) Then
        returninfo = returninfo + ",Baud Rate Response Command
Error"
    Else
        returninfo = returninfo + ",Baud Rate Fail"
        returninfoDlg = returninfoDlg + " " + setinfo + "Baud Rate
Fail Command Response=0x" + Convert.ToString(fCmdRet) + "(" +
GetReturnCodeDesc(fCmdRet) + ")"
    End If

    progressBar1.Value = 70
    fCmdRet = StaticClassReaderB.WriteScanTime(fComAdr,
scantime, frmcomportindex)
    If (fCmdRet = 0) Then
        returninfo = returninfo + ",InventoryScanTime Success"

```

```

ElseIf (fCmdRet = &HEE) Then
    returninfo = returninfo + ",InventoryScanTime Response
Command Error"
Else
    returninfo = returninfo + ",InventoryScanTime Fail"
    returninfoDlg = returninfoDlg + " " + setinfo +
"InventoryScanTime Fail Command Response=0x" +
Convert.ToString(fCmdRet) + "(" + GetReturnCodeDesc(fCmdRet) + ")"
End If

progressBar1.Value = 100
Button3_Click(sender, e)
progressBar1.Visible = False
StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + returninfo
If (returninfoDlg <> "") Then
    MessageBox.Show(returninfoDlg, "Information")
End If
End Sub

Private Sub CheckBox_SameFre_CheckedChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
CheckBox_SameFre.CheckedChanged
    If (CheckBox_SameFre.Checked) Then
        ComboBox_dmaxfre.SelectedIndex =
ComboBox_dminfre.SelectedIndex
    End If
End Sub

Private Sub ComboBox_dminfre_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs)
    If (CheckBox_SameFre.Checked) Then
        ComboBox_dminfre.SelectedIndex =
ComboBox_dmaxfre.SelectedIndex
    ElseIf (ComboBox_dminfre.SelectedIndex >
ComboBox_dmaxfre.SelectedIndex) Then

        ComboBox_dminfre.SelectedIndex =
ComboBox_dmaxfre.SelectedIndex

```

```

        MessageBox.Show("Min.Frequency is equal or lesser than
Max.Frequency", "Error Information")
    End If
End Sub
Public Sub ChangeSubItem(ByVal ListItem As ListViewItem,
ByVal subItemIndex As Integer, ByVal ItemText As String)
    If (subItemIndex = 1) Then
        If (ItemText = "") Then
            ListItem.SubItems(subItemIndex).Text = ItemText
            If (ListItem.SubItems(subItemIndex + 2).Text = "") Then
                ListItem.SubItems(subItemIndex + 2).Text = "1"
            Else
                ListItem.SubItems(subItemIndex + 2).Text =
Convert.ToString(Convert.ToInt32(ListItem.SubItems(subItemIndex +
2).Text) + 1)
            End If
        ElseIf (ListItem.SubItems(subItemIndex).Text <>
ItemText) Then
            ListItem.SubItems(subItemIndex).Text = ItemText
            ListItem.SubItems(subItemIndex + 2).Text = "1"
        Else
            ListItem.SubItems(subItemIndex + 2).Text =
Convert.ToString(Convert.ToInt32(ListItem.SubItems(subItemIndex +
2).Text) + 1)
            If ((Convert.ToUInt32(ListItem.SubItems(subItemIndex
+ 2).Text) > 9999)) Then
                ListItem.SubItems(subItemIndex + 2).Text = "1"
            End If
        End If
    ElseIf (subItemIndex = 2) Then
        If (ListItem.SubItems(subItemIndex).Text <> ItemText)
Then
            ListItem.SubItems(subItemIndex).Text = ItemText
        End If
    End If

End Sub

```

```

Private Sub button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button2.Click
    If (CheckBox_TID.Checked) Then
        If ((textBox4.Text.Length) <> 2 Or ((textBox5.Text.Length)
<> 2)) Then
            StatusBar1.Panels(0).Text = "Query TID Parameter
Error!;"
            Return
        End If
    End If
    Timer_Test_.Enabled = Not Timer_Test_.Enabled
    If (Not Timer_Test_.Enabled) Then
        textBox4.Enabled = True
        textBox5.Enabled = True
        CheckBox_TID.Enabled = True
        If (ListView1_EPC.Items.Count <> 0) Then
            DestroyCode.Enabled = False
            AccessCode.Enabled = False
            NoProect.Enabled = False
            Proect.Enabled = False
            Always.Enabled = False
            AlwaysNot.Enabled = False
            NoProect2.Enabled = True
            Proect2.Enabled = True
            Always2.Enabled = True
            AlwaysNot2.Enabled = True
            P_Reserve.Enabled = True
            P_EPC.Enabled = True
            P_TID.Enabled = True
            P_User.Enabled = True
            Button_DestroyCard.Enabled = True
            Button_SetReadProtect_G2.Enabled = True
            Button_SetEASAlarm_G2.Enabled = True
            Alarm_G2.Enabled = True
            NoAlarm_G2.Enabled = True
            Button_LockUserBlock_G2.Enabled = True
            Button_WriteEPC_G2.Enabled = True
            Button_SetMultiReadProtect_G2.Enabled = True
            Button_RemoveReadProtect_G2.Enabled = True
        End If
    End If
End Sub

```

```

Button_CheckReadProtected_G2.Enabled = True
button4.Enabled = True
SpeedButton_Read_G2.Enabled = True
Button_SetProtectState.Enabled = True
Button_DataWrite.Enabled = True
Button_BlockWrite.Enabled = True
Button_BlockErase.Enabled = True
Button_BlockWrite.Enabled = True
checkBox1.Enabled = True
End If
If (ListView1_EPC.Items.Count = 0) Then
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = False
    Proect2.Enabled = False
    Always2.Enabled = False
    AlwaysNot2.Enabled = False
    P_Reserve.Enabled = False
    P_EPC.Enabled = False
    P_TID.Enabled = False
    P_User.Enabled = False
    Button_DestroyCard.Enabled = False
    Button_SetReadProtect_G2.Enabled = False
    Button_SetEASAlarm_G2.Enabled = False
    Alarm_G2.Enabled = False
    NoAlarm_G2.Enabled = False
    Button_LockUserBlock_G2.Enabled = False
    SpeedButton_Read_G2.Enabled = False
    Button_DataWrite.Enabled = False
    Button_BlockWrite.Enabled = False
    Button_BlockErase.Enabled = False
    Button_WriteEPC_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True

```

```
button4.Enabled = True
Button_SetProtectState.Enabled = False
checkBox1.Enabled = False
```

End If

```
AddCmdLog("Inventory", "Exit Query", 0)
button2.Text = "Query Tag"
```

Else

```
textBox4.Enabled = False
textBox5.Enabled = False
CheckBox_TID.Enabled = False
DestroyCode.Enabled = False
AccessCode.Enabled = False
NoProect.Enabled = False
Proect.Enabled = False
Always.Enabled = False
AlwaysNot.Enabled = False
NoProect2.Enabled = False
Proect2.Enabled = False
Always2.Enabled = False
AlwaysNot2.Enabled = False
P_Reserve.Enabled = False
P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Button_WriteEPC_G2.Enabled = False
Button_SetMultiReadProtect_G2.Enabled = False
Button_RemoveReadProtect_G2.Enabled = False
Button_CheckReadProtected_G2.Enabled = False
button4.Enabled = False
```

```
Button_DestroyCard.Enabled = False
Button_SetReadProtect_G2.Enabled = False
Button_SetEASAlarm_G2.Enabled = False
Alarm_G2.Enabled = False
NoAlarm_G2.Enabled = False
Button_LockUserBlock_G2.Enabled = False
SpeedButton_Read_G2.Enabled = False
Button_DataWrite.Enabled = False
```

```

        Button_BlockWrite.Enabled = False
        Button_BlockErase.Enabled = False
        Button_SetProtectState.Enabled = False
        ListView1_EPC.Items.Clear()
        ComboBox_EPC1.Items.Clear()
        ComboBox_EPC2.Items.Clear()
        ComboBox_EPC3.Items.Clear()
        ComboBox_EPC4.Items.Clear()
        ComboBox_EPC5.Items.Clear()
        ComboBox_EPC6.Items.Clear()
        button2.Text = "Stop"
        checkBox1.Enabled = False
    End If
End Sub
Private Sub Inventory()
    Dim i As Integer
    Dim CardNum As Integer = 0
    Dim Totallen As Integer = 0
    Dim EPCLen, m As Integer
    Dim EPC(5000) As Byte
    Dim CardIndex As Integer
    Dim temps As String
    Dim s, sEPC As String
    Dim isonlistview As Boolean
    Dim aListItem As ListViewItem
    Dim AdrTID As Byte = 0
    Dim LenTID As Byte = 0
    Dim TIDFlag As Byte = 0
    If (CheckBox_TID.Checked) Then
        AdrTID = Convert.ToByte(textBox4.Text, 16)
        LenTID = Convert.ToByte(textBox5.Text, 16)
        TIDFlag = 1
    Else
        AdrTID = 0
        LenTID = 0
        TIDFlag = 0
    End If
    fCmdRet = StaticClassReaderB.Inventory_G2(fComAdr,
    AdrTID, LenTID, TIDFlag, EPC, Totallen, CardNum, frmcomportindex)

```

```

        If ((fCmdRet = 1) Or (fCmdRet = 2) Or (fCmdRet = 3) Or
(fCmdRet = 4) Or (fCmdRet = &HFB)) Then
            Dim daw(Totallen) As Byte
            Array.Copy(EPC, daw, Totallen)
            temps = ByteArrayToHexString(daw)
            flInventory_EPC_List = temps
            m = 0

            If (CardNum = 0) Then
                flInventoryScan = False
                Exit Sub
            End If
            For CardIndex = 0 To CardNum - 1
                EPClen = daw(m)
                sEPC = temps.Substring(m * 2 + 2, EPClen * 2)
                m = m + EPClen + 1
                If (sEPC.Length <> EPClen * 2) Then
                    flInventoryScan = False
                    Exit Sub
                End If
                isonlistview = False
                For i = 0 To ListView1_EPC.Items.Count - 1
                    If (sEPC = ListView1_EPC.Items(i).SubItems(1).Text)
Then
                        aListItem = ListView1_EPC.Items(i)
                        ChangeSubItem(aListItem, 1, sEPC)
                        isonlistview = True
                    End If
                Next i
                If (Not isonlistview) Then
                    aListItem
                    ListView1_EPC.Items.Add((ListView1_EPC.Items.Count
1).ToString())
                    aListItem.SubItems.Add("")
                    aListItem.SubItems.Add("")
                    aListItem.SubItems.Add("")
                    s = sEPC
                    ChangeSubItem(aListItem, 1, s)
                End If
            Next i
        End If
    End Sub

```



```

        s = (sEPC.Length / 2).ToString().PadLeft(2, "0")
        ChangeSubItem(aListItem, 2, s)
        If (Not CheckBox_TID.Checked) Then
            If (ComboBox_EPC1.Items.IndexOf(sEPC) = -1)
                Then
                    ComboBox_EPC1.Items.Add(sEPC)
                    ComboBox_EPC2.Items.Add(sEPC)
                    ComboBox_EPC3.Items.Add(sEPC)
                    ComboBox_EPC4.Items.Add(sEPC)
                    ComboBox_EPC5.Items.Add(sEPC)
                    ComboBox_EPC6.Items.Add(sEPC)
                    TextBox6.Text = sEPC
                End If
            End If
        End If
    Next CardIndex
End If
If (Not CheckBox_TID.Checked) Then
    If ((ComboBox_EPC1.Items.Count <> 0)) Then
        ComboBox_EPC1.SelectedIndex = 0
        ComboBox_EPC2.SelectedIndex = 0
        ComboBox_EPC3.SelectedIndex = 0
        ComboBox_EPC4.SelectedIndex = 0
        ComboBox_EPC5.SelectedIndex = 0
        ComboBox_EPC6.SelectedIndex = 0
    End If
End If
If (fAppClosed) Then
    Close()
End If
TextBox10.Text = TextBox6.Text
End Sub

Private Sub Timer_Test__Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer_Test__Tick
    If (flsInventoryScan) Then
        Exit Sub
    End If
    flsInventoryScan = True

```

```

Inventory()
flsInventoryScan = False
End Sub

Private Sub SpeedButton_Read_G2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SpeedButton_Read_G2.Click
    If (Edit_WordPtr.Text = "") Then
        MessageBox.Show("Address of Tag Data is NULL",
"Information")
        Exit Sub
    End If
    If (textBox1.Text = "") Then

        MessageBox.Show("Length of Data(Read/Block Erase) is
NULL", "Information")
        Exit Sub
    End If
    If (Edit_AccessCode2.Text = "") Then
        MessageBox.Show("(PassWord) is NULL", "Information")
        Exit Sub
    End If
    If (Convert.ToInt32(Edit_WordPtr.Text, 16) +
Convert.ToInt32(textBox1.Text) > 120) Then
        Exit Sub
    End If
    Timer_G2_Read.Enabled = Not Timer_G2_Read.Enabled
    If (Timer_G2_Read.Enabled) Then
        DestroyCode.Enabled = False
        AccessCode.Enabled = False
        NoProect.Enabled = False
        Proect.Enabled = False
        Always.Enabled = False
        AlwaysNot.Enabled = False
        NoProect2.Enabled = False
        Proect2.Enabled = False
        Always2.Enabled = False
        AlwaysNot2.Enabled = False
        P_Reserve.Enabled = False
    End If
End Sub

```

```

P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Button_WriteEPC_G2.Enabled = False
Button_SetMultiReadProtect_G2.Enabled = False
Button_RemoveReadProtect_G2.Enabled = False
Button_CheckReadProtected_G2.Enabled = False
button4.Enabled = False

Button_DestroyCard.Enabled = False
Button_SetReadProtect_G2.Enabled = False
Button_SetEASAlarm_G2.Enabled = False
Alarm_G2.Enabled = False
NoAlarm_G2.Enabled = False
Button_LockUserBlock_G2.Enabled = False
button2.Enabled = False
Button_DataWrite.Enabled = False
Button_BlockWrite.Enabled = False
Button_BlockErase.Enabled = False
Button_SetProtectState.Enabled = False
SpeedButton_Read_G2.Text = "Stop"
Else
If (ListView1_EPC.Items.Count <> 0) Then
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = True
    Proect2.Enabled = True
    Always2.Enabled = True
    AlwaysNot2.Enabled = True
    P_Reserve.Enabled = True
    P_EPC.Enabled = True
    P_TID.Enabled = True
    P_User.Enabled = True
    Button_DestroyCard.Enabled = True
    Button_SetReadProtect_G2.Enabled = True

```

```

Button_SetEASAlarm_G2.Enabled = True
Alarm_G2.Enabled = True
NoAlarm_G2.Enabled = True
Button_LockUserBlock_G2.Enabled = True
Button_WriteEPC_G2.Enabled = True
Button_SetMultiReadProtect_G2.Enabled = True
Button_RemoveReadProtect_G2.Enabled = True
Button_CheckReadProtected_G2.Enabled = True
button4.Enabled = True
button2.Enabled = True
Button_SetProtectState.Enabled = True

Button_DataWrite.Enabled = True
Button_BlockWrite.Enabled = True
Button_BlockErase.Enabled = True
End If
If (ListView1_EPC.Items.Count = 0) Then
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = False
    Proect2.Enabled = False
    Always2.Enabled = False
    AlwaysNot2.Enabled = False
    P_Reserve.Enabled = False
    P_EPC.Enabled = False
    P_TID.Enabled = False
    P_User.Enabled = False
    Button_DestroyCard.Enabled = False
    Button_SetReadProtect_G2.Enabled = False
    Button_SetEASAlarm_G2.Enabled = False
    Alarm_G2.Enabled = False
    NoAlarm_G2.Enabled = False
    Button_LockUserBlock_G2.Enabled = False
    Button_SetProtectState.Enabled = False
    button2.Enabled = True

```

```

        Button_DataWrite.Enabled = False
        Button_BlockWrite.Enabled = False
        Button_BlockErase.Enabled = False
        Button_WriteEPC_G2.Enabled = True
        Button_SetMultiReadProtect_G2.Enabled = True
        Button_RemoveReadProtect_G2.Enabled = True
        Button_CheckReadProtected_G2.Enabled = True
        button4.Enabled = True
    End If
    SpeedButton_Read_G2.Text = "Read"
End If

End Sub

Private Sub Timer_G2_Read_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer_G2_Read.Tick
    If (flsInventoryScan) Then
        Exit Sub
    End If
    flsInventoryScan = True
    Dim WordPtr, ENum1 As Byte
    Dim Num As Byte = 0
    Dim Mem As Byte = 0
    Dim EPCLength As Byte = 0
    Dim str As String
    Dim CardData(320) As Byte
    If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
        flsInventoryScan = False
        Exit Sub
    End If
    If (checkBox1.Checked) Then
        MaskFlag = 1
    Else
        MaskFlag = 0
    End If
    Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
    MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)

```

```

If (textBox1.Text = "") Then
    flsInventoryScan = False
    Exit Sub
End If
If (ComboBox_EPC2.Items.Count = 0) Then
    flsInventoryScan = False
    Exit Sub
End If
If (ComboBox_EPC2.SelectedItem = Nothing) Then
    flsInventoryScan = False
    Exit Sub
End If
str = ComboBox_EPC2.SelectedItem.ToString()
ENum1 = Convert.ToByte(str.Length / 4)
EPClength = Convert.ToByte(str.Length / 2)
Dim EPC(ENum1) As Byte
EPC = HexStringToByteArray(str)
If (C_Reserve.Checked) Then
    Mem = 0
End If
If (C_EPC.Checked) Then
    Mem = 1
End If
If (C_TID.Checked) Then
    Mem = 2
End If
If (C_User.Checked) Then
    Mem = 3
End If
If (Edit_AccessCode2.Text = "") Then
    flsInventoryScan = False
    Exit Sub
End If
If (Edit_WordPtr.Text = "") Then
    flsInventoryScan = False
    Exit Sub
End If
WordPtr = Convert.ToByte(Edit_WordPtr.Text, 16)
Num = Convert.ToByte(textBox1.Text)

```

```

    If (Edit_AccessCode2.Text.Length <> 8) Then
        flsInventoryScan = False
        Exit Sub
    End If
    fPassWord =
HexStringToByteArray(Edit_AccessCode2.Text)
    fCmdRet = StaticClassReaderB.ReadCard_G2(fComAdr,
EPC, Mem, WordPtr, Num, fPassWord, Maskadr, MaskLen, MaskFlag,
CardData, EPCLength, ferrorcode, frmcomportindex)
    If (fCmdRet = 0) Then
        Dim daw(Num * 2 - 1) As Byte
        Array.Copy(CardData, daw, Num * 2)
        listBox1.Items.Add(ByteArrayToHexString(daw))
        listBox1.SelectedIndex = listBox1.Items.Count - 1
        AddCmdLog("ReadData", "Read", fCmdRet)
    End If
    If (ferrorcode <> -1) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Read' Response
Error=0x" + Convert.ToString(ferrorcode, 2) + "(" +
GetErrorDesc(ferrorcode) + ")"
        ferrorcode = -1
    End If
    flsInventoryScan = False
    If (fAppClosed) Then
        Close()
    End If
End Sub

Private Sub Button_DataWrite_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_DataWrite.Click
    Dim WordPtr, ENum1 As Byte
    Dim Num As Byte = 0
    Dim Mem As Byte = 0
    Dim WNum As Byte = 0
    Dim EPCLength As Byte = 0
    Dim Writedatalen As Byte = 0
    Dim WrittenDataNum As Integer = 0

```

```

Dim s2, str As String
Dim CardData(320) As Byte
Dim writedata(230) As Byte
If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
    Exit Sub
End If
If (checkBox1.Checked) Then
    MaskFlag = 1
Else
    MaskFlag = 0
End If
Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
If (ComboBox_EPC2.Items.Count = 0) Then
    Exit Sub
End If
If (ComboBox_EPC2.SelectedItem = Nothing) Then
    Exit Sub
End If
str = ComboBox_EPC2.SelectedItem.ToString()
ENum1 = Convert.ToByte(str.Length / 4)
EPClength = Convert.ToByte(ENum1 * 2)
Dim EPC(ENum1) As Byte
EPC = HexStringToByteArray(str)
If (C_Reserve.Checked) Then
    Mem = 0
End If
If (C_EPC.Checked) Then
    Mem = 1
End If
If (C_TID.Checked) Then
    Mem = 2
End If
If (C_User.Checked) Then
    Mem = 3
End If
If (Edit_WordPtr.Text = "") Then

```



```

        MessageBox.Show("Address of Tag Data is NULL",
"Information")
        Exit Sub
    End If
    If (textBox1.Text = "") Then
        MessageBox.Show("Length of Data(Read/Block Erase) is
NULL", "Information")
        Exit Sub
    End If
    If (Convert.ToInt32(Edit_WordPtr.Text, 16) +
Convert.ToInt32(textBox1.Text) > 120) Then
        Exit Sub
    End If
    If (Edit_AccessCode2.Text = "") Then
        Exit Sub
    End If
    WordPtr = Convert.ToByte(Edit_WordPtr.Text, 16)
    Num = Convert.ToByte(textBox1.Text)
    If (Edit_AccessCode2.Text.Length <> 8) Then
        Exit Sub
    End If
    fPassWord =
HexStringToByteArray(Edit_AccessCode2.Text)
    If (Edit_WriteData.Text = "") Then
        Exit Sub
    End If
    s2 = Edit_WriteData.Text
    If (s2.Length Mod 4 <> 0) Then
        MessageBox.Show("The Number must be 4 times.",
"Wtite")
        Exit Sub
    End If
    WNum = Convert.ToByte(s2.Length / 4)
    ReDim writedata(WNum * 2)
    writedata = HexStringToByteArray(s2)
    Writedatalen = Convert.ToByte(WNum * 2)
    If ((checkBox_pc.Checked) And (C_EPC.Checked)) Then
        WordPtr = 1

```

```

        Writedatalen =
Convert.ToByte(Edit_WriteData.Text.Length / 2 + 2)
        writedata = HexStringToByteArray(textBox_pc.Text +
Edit_WriteData.Text)
    End If
    fCmdRet = StaticClassReaderB.WriteCard_G2(fComAdr,
EPC, Mem, WordPtr, Writedatalen, writedata, fPassWord, Maskadr,
MaskLen, MaskFlag, WrittenDataNum, EPCLength, ferrortcode,
frmcomportindex)
    AddCmdLog("Write data", "write", fCmdRet)
    If (fCmdRet = 0) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + ";@'Write'Command
Response=0x00'(completely write Data successfully)"
    End If
End Sub

Private Sub Button_BlockErase_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_BlockErase.Click
    Dim WordPtr, ENum1 As Byte
    Dim Num As Byte = 0
    Dim Mem As Byte = 0
    Dim EPCLength As Byte = 0
    Dim str As String
    Dim CardData(320) As Byte
    If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
        Exit Sub
    End If
    If (checkBox1.Checked) Then
        MaskFlag = 1
    Else
        MaskFlag = 0
    End If
    Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
    MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
    If (ComboBox_EPC2.Items.Count = 0) Then
        Exit Sub
    End If

```

```

End If
If (ComboBox_EPC2.SelectedItem = Nothing) Then
    Exit Sub
End If
str = ComboBox_EPC2.SelectedItem.ToString()
If (str = "") Then
    Exit Sub
End If
ENum1 = Convert.ToByte(str.Length / 4)
EPClength = Convert.ToByte(str.Length / 2)
Dim EPC(ENum1) As Byte
EPC = HexStringToByteArray(str)
If (C_Reserve.Checked) Then
    Mem = 0
End If
If (C_EPC.Checked) Then
    Mem = 1
End If
If (C_TID.Checked) Then
    Mem = 2
End If
If (C_User.Checked) Then
    Mem = 3
End If
If (Edit_WordPtr.Text = "") Then
    MessageBox.Show("Address of Tag Data is NULL",
"Information")
    Exit Sub
End If
If (textBox1.Text = "") Then
    MessageBox.Show("Length of Data(Read/Block Erase) is
NULL", "Information")
    Exit Sub
End If
If (Convert.ToInt32(Edit_WordPtr.Text, 16) +
Convert.ToInt32(textBox1.Text) > 120) Then
    Exit Sub
End If
If (Edit_AccessCode2.Text = "") Then

```

```

        Exit Sub
    End If
    WordPtr = Convert.ToByte(Edit_WordPtr.Text, 16)
    If ((Mem = 1) And (WordPtr < 2)) Then
        MessageBox.Show("the length of start Address of erasing
EPC area is equal or greater than 0x01!", "Information")
    End Sub
    End If
    Num = Convert.ToByte(textBox1.Text)
    If (Edit_AccessCode2.Text.Length <> 8) Then
        Exit Sub
    End If
    fPassWord =
HexStringToByteArray(Edit_AccessCode2.Text)
    fCmdRet = StaticClassReaderB.EraseCard_G2(fComAdr,
EPC, Mem, WordPtr, Num, fPassWord, Maskadr, MaskLen, MaskFlag,
EPClength, ferrorcode, frmcomportindex)
    AddCmdLog("EraseCard", "Erase data", fCmdRet)
    If (fCmdRet = 0) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Block Erase'Command
Response=0x00" + "(Block Erase successfully)"
    End If
End Sub

Private Sub button7_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button7.Click
    listBox1.Items.Clear()
End Sub

Private Sub Button_SetProtectState_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_SetProtectState.Click
    Dim select1 As Byte = 0
    Dim setprotect As Byte = 0
    Dim EPClength As Byte
    Dim Str As String
    Dim ENum1 As Byte

```

```

        If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
            flsInventoryScan = False
            Exit Sub
        End If
        If (checkBox1.Checked) Then
            MaskFlag = 1
        Else
            MaskFlag = 0
        End If
        Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
        MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
        If (ComboBox_EPC1.Items.Count = 0) Then
            Exit Sub
        End If
        If (ComboBox_EPC1.SelectedItem = Nothing) Then
            Exit Sub
        End If
        Str = ComboBox_EPC1.SelectedItem.ToString()
        If (Str = "") Then
            Exit Sub
        End If
        Enum1 = Convert.ToByte(Str.Length / 4)
        Dim EPC(200) As Byte
        EPC = HexStringToByteArray(Str)
        EPClength = Str.Length() / 2
        If (textBox2.Text.Length <> 8) Then
            MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
            Exit Sub
        End If
        fPassWord = HexStringToByteArray(textBox2.Text)
        If ((P_Reserve.Checked) And (DestroyCode.Checked)) Then
            select1 = &H0
        ElseIf ((P_Reserve.Checked) And (AccessCode.Checked))
Then
            select1 = &H1
        ElseIf (P_EPC.Checked) Then
            select1 = &H2

```

```

ElseIf (P_TID.Checked) Then
    select1 = &H3
ElseIf (P_User.Checked) Then
    select1 = &H4
End If
If (P_Reserve.Checked) Then
    If (NoProect.Checked) Then
        setprotect = &H0
    ElseIf (Proect.Checked) Then
        setprotect = &H2
    ElseIf (Always.Checked) Then
        setprotect = &H1
        If (MessageBox.Show(Me, "Set readable and writeable
Confirmed?", "Information", MessageBoxButtons.OKCancel) =
DialogResult.Cancel) Then
            Exit Sub
        End If
    ElseIf (AlwaysNot.Checked) Then
        setprotect = &H3
        If (MessageBox.Show(Me, "Set never readable and
writeable Confirmed?", "Information", MessageBoxButtons.OKCancel)
= DialogResult.Cancel) Then
            Exit Sub
        End If
    End If
End If
Else
    If (NoProect2.Checked) Then
        setprotect = &H0
    ElseIf (Proect2.Checked) Then
        setprotect = &H2
    ElseIf (Always2.Checked) Then
        setprotect = &H1
        If (MessageBox.Show(Me, "Set writeable Confirmed?",
"Information", MessageBoxButtons.OKCancel) = DialogResult.Cancel)
Then
            Exit Sub
        End If
    ElseIf (AlwaysNot2.Checked) Then

```

```

        setprotect = &H3
        If (MessageBox.Show(Me, "Set never writeable
Confirmed?", "Information", MessageBoxButtons.OKCancel) =
DialogResult.Cancel) Then
            Exit Sub
        End If
    End If
End If
fCmdRet =
StaticClassReaderB.SetCardProtect_G2(fComAdr, EPC, select1,
setprotect, fPassWord, Maskadr, MaskLen, MaskFlag, EPClength,
ferrorcode, frmcomportindex)
AddCmdLog("SetCardProtect", "SetProtect", fCmdRet)
End Sub

Private Sub P_Reserve_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles P_Reserve.Click
    If (ListView1_EPC.Items.Count <> 0) Then
        DestroyCode.Enabled = True
        AccessCode.Enabled = True
        NoProect.Enabled = True
        Proect.Enabled = True
        Always.Enabled = True
        AlwaysNot.Enabled = True
        NoProect2.Enabled = False
        Proect2.Enabled = False
        Always2.Enabled = False
        AlwaysNot2.Enabled = False
    End If
End Sub

Private Sub P_EPC_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles P_EPC.Click
    If (ListView1_EPC.Items.Count <> 0) Then
        DestroyCode.Enabled = False
        AccessCode.Enabled = False
        NoProect.Enabled = False
        Proect.Enabled = False
        Always.Enabled = False
    End If
End Sub

```

```

        AlwaysNot.Enabled = False
        NoProect2.Enabled = True
        Proect2.Enabled = True
        Always2.Enabled = True
        AlwaysNot2.Enabled = True
    End If
End Sub

```

```

Private Sub P_TID_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles P_TID.Click
    If (ListView1_EPC.Items.Count <> 0) Then
        DestroyCode.Enabled = False
        AccessCode.Enabled = False
        NoProect.Enabled = False
        Proect.Enabled = False
        Always.Enabled = False
        AlwaysNot.Enabled = False
        NoProect2.Enabled = True
        Proect2.Enabled = True
        Always2.Enabled = True
        AlwaysNot2.Enabled = True
    End If
End Sub

```

```

Private Sub P_User_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles P_User.Click
    If (ListView1_EPC.Items.Count <> 0) Then
        DestroyCode.Enabled = False
        AccessCode.Enabled = False
        NoProect.Enabled = False
        Proect.Enabled = False
        Always.Enabled = False
        AlwaysNot.Enabled = False
        NoProect2.Enabled = True
        Proect2.Enabled = True
        Always2.Enabled = True
        AlwaysNot2.Enabled = True
    End If
End Sub

```



```

        Private Sub Button_DestroyCard_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_DestroyCard.Click
    Dim EPCLength As Byte
    Dim str As String
    Dim ENum1 As Byte
    If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
        Exit Sub
    End If
    If (checkBox1.Checked) Then
        MaskFlag = 1
    Else
        MaskFlag = 0
    End If
    Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
    MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
    StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + ""
    If (Edit_DestroyCode.Text.Length <> 8) Then
        MessageBox.Show("Kill Password Less Than 8 digit!Please
input again!", "Information")
        Exit Sub
    End If
    If (ComboBox_EPC3.Items.Count = 0) Then
        Exit Sub
    End If
    If (ComboBox_EPC3.SelectedItem = Nothing) Then
        Exit Sub
    End If
    str = ComboBox_EPC3.SelectedItem.ToString()
    If (str = "") Then
        Exit Sub
    End If
    If (MessageBox.Show(Me, "Kill the Tag Confirmed?",
"Information", MessageBoxButtons.OKCancel) = DialogResult.Cancel)
Then
        Exit Sub

```

```

End If
Enum1 = Convert.ToByte(str.Length / 4)
EPCLength = Convert.ToByte(str.Length / 2)
Dim EPC(Enum1) As Byte
EPC = HexStringToByteArray(str)
fPassWord =
HexStringToByteArray(Edit_DestroyCode.Text)
fCmdRet = StaticClassReaderB.DestroyCard_G2(fComAdr,
EPC, fPassWord, MaskAdr, MaskLen, MaskFlag, EPCLength, ferrrcode,
frmcomportindex)
AddCmdLog("DestroyCard", "Kill Tag", fCmdRet)
If (fCmdRet = 0) Then
    StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Kill Tag'Command
Response=0x00" + "(Kill successfully)"
End If
End Sub

Private Sub Button_WriteEPC_G2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_WriteEPC_G2.Click
    Dim WriteEPC(100) As Byte
    Dim WriteEPClen As Byte
    Dim Enum1 As Byte
    If (Edit_AccessCode3.Text.Length < 8) Then
        MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
    Exit Sub
    End If
    If ((Edit_WriteEPC.Text.Length Mod 4) <> 0) Then

        MessageBox.Show("Please input Data in words in
hexadecimal form!", "Information")
    Exit Sub
    End If
    WriteEPClen = Convert.ToByte(Edit_WriteEPC.Text.Length
/ 2)
    Enum1 = Convert.ToByte(Edit_WriteEPC.Text.Length / 4)
    Dim EPC(Enum1) As Byte

```

```

        EPC = HexStringToByteArray(Edit_WriteEPC.Text)
        fPassWord = HexStringToByteArray(Edit_AccessCode3.Text)
        fCmdRet = StaticClassReaderB.WriteEPC_G2(fComAdr,
        fPassWord, EPC, WriteEPClen, ferrrcode, frmcomportindex)
        AddCmdLog("WriteEPC_G2", "Write EPC", fCmdRet)
        If (fCmdRet = 0) Then
            StatusBar1.Panels(0).Text =
            DateTime.Now.ToLongTimeString() + " 'Write EPC'Command
            Response=0x00" + "(Write EPC successfully)"
        End If
    End Sub

```

```

    Private Sub Button_SetReadProtect_G2_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button_SetReadProtect_G2.Click
        Dim EPClength As Byte
        Dim ENum1 As Byte
        Dim str As String
        If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
        "")) Then
            Exit Sub
        End If
        If (checkBox1.Checked) Then
            MaskFlag = 1
        Else
            MaskFlag = 0
        End If
        Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
        MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
        If (Edit_AccessCode4.Text.Length < 8) Then
            MessageBox.Show("Access Password Less Than 8
            digit!Please input again!", "Information")
        End If
        Exit Sub
    End If
    If (ComboBox_EPC4.Items.Count = 0) Then
        Exit Sub
    End If
    If (ComboBox_EPC4.SelectedItem = Nothing) Then

```

```

Exit Sub
End If
str = ComboBox_EPC4.SelectedItem.ToString()
If (str = "") Then
Exit Sub
End If
ENum1 = Convert.ToByte(str.Length / 4)
EPClength = Convert.ToByte(str.Length / 2)
Dim EPC(ENum1) As Byte
EPC = HexStringToByteArray(str)
fPassWord =
HexStringToByteArray(Edit_AccessCode4.Text)
fCmdRet =
StaticClassReaderB.SetReadProtect_G2(fComAdr, EPC, fPassWord,
MaskAdr, MaskLen, MaskFlag, EPClength, ferrorcode,
frmcomportindex)
AddCmdLog("SetReadProtect_G2", "Set Single Tag Read
Protection", fCmdRet)
If (fCmdRet = 0) Then
StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Set Single Tag Read
Protection'Command Response=0x00" + "Set Single Tag Read Protection
successfully"
End If
End Sub

Private Sub Button_SetMultiReadProtect_G2_Click(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
Button_SetMultiReadProtect_G2.Click
If (Edit_AccessCode4.Text.Length < 8) Then
MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
Exit Sub
End If
fPassWord =
HexStringToByteArray(Edit_AccessCode4.Text)
fCmdRet =
StaticClassReaderB.SetMultiReadProtect_G2(fComAdr, fPassWord,
ferrorcode, frmcomportindex)

```

```

        AddCmdLog("SetMultiReadProtect_G2", "Set Single Tag
Read Protection without EPC", fCmdRet)
        If (fCmdRet = 0) Then
            StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Set Single Tag Read Protection
without EPC'Command Response=0x00" + "(Set Single Tag Read
Protection without EPC successfully)"
        End If
    End Sub

```

```

    Private Sub Button_RemoveReadProtect_G2_Click(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
Button_RemoveReadProtect_G2.Click

```

```

        If (Edit_AccessCode4.Text.Length < 8) Then
            MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
        End If
    End Sub

```

```

        fPassWord =
HexStringToByteArray(Edit_AccessCode4.Text)
        fCmdRet =
StaticClassReaderB.RemoveReadProtect_G2(fComAdr, fPassWord,
ferrorcode, frmcomportindex)

```

```

        AddCmdLog("RemoveReadProtect_G2", "Reset Single Tag
Read Protection", fCmdRet)

```

```

        If (fCmdRet = 0) Then
            StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Reset Single Tag Read
Protection'Command Response=0x00" + "(Reset Single Tag Read
Protection successfully)"
        End If
    End Sub

```

```

    Private Sub Button_CheckReadProtected_G2_Click(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
Button_CheckReadProtected_G2.Click

```

```

        Dim readpro As Byte = 2

```

```

        fCmdRet =
StaticClassReaderB.CheckReadProtected_G2(fComAdr, readpro,
ferrorcode, frmcomportindex)
        AddCmdLog("CheckReadProtected_G2", "Detect Single Tag
Read Protection", fCmdRet)
        If (fCmdRet = 0) Then
            If (readpro = 0) Then
                StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Detect Single Tag Read
Protection'Command Response=0x00" + "(Single Tag is unprotected)"
            End If
            If (readpro = 1) Then
                StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Detect Single Tag Read
Protection'Command Response=0x00" + "(Single Tag is protected)"
            End If
        End If
    End Sub

    Private Sub Button_SetEASAlarm_G2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_SetEASAlarm_G2.Click
        Dim EPCLength As Byte = 0
        Dim EAS As Byte = 0
        Dim ENum1 As Byte
        Dim str As String
        If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
            Exit Sub
        End If
        If (checkBox1.Checked) Then
            MaskFlag = 1
        Else
            MaskFlag = 0
        End If
        Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
        MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
        If (Edit_AccessCode5.Text.Length < 8) Then

```

```

        MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
    Exit Sub
End If
If (ComboBox_EPC5.Items.Count = 0) Then
    Exit Sub
End If
If (ComboBox_EPC5.SelectedItem = Nothing) Then
    Exit Sub
End If
str = ComboBox_EPC5.SelectedItem.ToString()
If (str = "") Then
    Exit Sub
End If
Enum1 = Convert.ToByte(str.Length / 4)
EPClength = Convert.ToByte(str.Length / 2)
Dim EPC(Enum1) As Byte
EPC = HexStringToByteArray(str)
fPassWord =
HexStringToByteArray(Edit_AccessCode5.Text)
If (Alarm_G2.Checked) Then
    EAS = 1
Else
    EAS = 0
End If
fCmdRet = StaticClassReaderB.SetEASAlarm_G2(fComAdr,
EPC, fPassWord, Maskadr, MaskLen, MaskFlag, EAS, EPClength,
ferrorcode, frmcomportindex)
AddCmdLog("SetEASAlarm_G2", "Alarm Setting",
fCmdRet)
If (fCmdRet = 0) Then
    If (Alarm_G2.Checked) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Alarm Setting'Command
Response=0x00" + "(Set EAS Alarm successfully)"
    Else
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Alarm Setting'Command
Response=0x00" + "(Clear EAS Alarm successfully)"
    End If
End If

```

```
End If
End If
End Sub
```

```
Private Sub button4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button4.Click
    Timer_G2_Alarm.Enabled = Not Timer_G2_Alarm.Enabled
    If (Timer_G2_Alarm.Enabled) Then
        DestroyCode.Enabled = False
        AccessCode.Enabled = False
        NoProect.Enabled = False
        Proect.Enabled = False
        Always.Enabled = False
        AlwaysNot.Enabled = False
        NoProect2.Enabled = False
        Proect2.Enabled = False
        Always2.Enabled = False
        AlwaysNot2.Enabled = False
        P_Reserve.Enabled = False
        P_EPC.Enabled = False
        P_TID.Enabled = False
        P_User.Enabled = False
        Button_WriteEPC_G2.Enabled = False
        Button_SetMultiReadProtect_G2.Enabled = False
        Button_RemoveReadProtect_G2.Enabled = False
        Button_CheckReadProtected_G2.Enabled = False
        button2.Enabled = False

        Button_DestroyCard.Enabled = False
        Button_SetReadProtect_G2.Enabled = False
        Button_SetEASAlarm_G2.Enabled = False
        Alarm_G2.Enabled = False
        NoAlarm_G2.Enabled = False
        Button_LockUserBlock_G2.Enabled = False
        SpeedButton_Read_G2.Enabled = False
        Button_DataWrite.Enabled = False
        Button_BlockWrite.Enabled = False
        Button_BlockErase.Enabled = False
        Button_SetProtectState.Enabled = False
```



```

button4.Text = "Stop"
Else
If (ListView1_EPC.Items.Count <> 0) Then
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = True
    Proect2.Enabled = True
    Always2.Enabled = True
    AlwaysNot2.Enabled = True
    P_Reserve.Enabled = True
    P_EPC.Enabled = True
    P_TID.Enabled = True
    P_User.Enabled = True
    Button_DestroyCard.Enabled = True
    Button_SetReadProtect_G2.Enabled = True
    Button_SetEASAlarm_G2.Enabled = True
    Alarm_G2.Enabled = True
    NoAlarm_G2.Enabled = True
    Button_LockUserBlock_G2.Enabled = True
    Button_WriteEPC_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True
    button2.Enabled = True
    Button_SetProtectState.Enabled = True
    SpeedButton_Read_G2.Enabled = True
    Button_DataWrite.Enabled = True
    Button_BlockWrite.Enabled = True
    Button_BlockErase.Enabled = True
End If
If (ListView1_EPC.Items.Count = 0) Then
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False

```

```

Always.Enabled = False
AlwaysNot.Enabled = False
NoProect2.Enabled = False
Proect2.Enabled = False
Always2.Enabled = False
AlwaysNot2.Enabled = False
P_Reserve.Enabled = False
P_EPC.Enabled = False
P_TID.Enabled = False
P_User.Enabled = False
Button_DestroyCard.Enabled = False
Button_SetReadProtect_G2.Enabled = False
Button_SetEASAlarm_G2.Enabled = False
Alarm_G2.Enabled = False
NoAlarm_G2.Enabled = False
Button_LockUserBlock_G2.Enabled = False
SpeedButton_Read_G2.Enabled = False
Button_DataWrite.Enabled = False
Button_BlockWrite.Enabled = False
Button_BlockErase.Enabled = False
Button_SetProtectState.Enabled = False
Button_WriteEPC_G2.Enabled = True
Button_SetMultiReadProtect_G2.Enabled = True
Button_RemoveReadProtect_G2.Enabled = True
Button_CheckReadProtected_G2.Enabled = True
button2.Enabled = True
End If
button4.Text = "Check Alarm"
Label_Alarm.Visible = False
StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Check EAS Alarm'over"
End If
End Sub

Private Sub Timer_G2_Alarm_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer_G2_Alarm.Tick
If (flsInventoryScan) Then
Exit Sub

```

```

        End If
        flsInventoryScan = True
        fCmdRet =
StaticClassReaderB.CheckEASAlarm_G2(fComAdr, ferrortcode,
frmcomportindex)
        If (fCmdRet = 0) Then
            StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Check EAS Alarm'Command
Response=0x00" + "(EAS alarm detected)"
            Label_Alarm.Visible = True
        Else
            Label_Alarm.Visible = False
            AddCmdLog("CheckEASAlarm_G2", "Check EAS
Alarm", fCmdRet)
        End If
        flsInventoryScan = False
        If (fAppClosed) Then
            Close()
        End If
    End Sub

    Private Sub Button_LockUserBlock_G2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button_LockUserBlock_G2.Click
        Dim EPCLength As Byte = 0
        Dim BlockNum As Byte = 0
        Dim ENum1 As Byte
        Dim str As String
        If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
            Exit Sub
        End If
        If (checkBox1.Checked) Then
            MaskFlag = 1
        Else
            MaskFlag = 0
        End If
        Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
        MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)

```

```

        If (Edit_AccessCode6.Text.Length < 8) Then
            MessageBox.Show("Access Password Less Than 8
digit!Please input again!", "Information")
            Exit Sub
        End If
        If (ComboBox_EPC6.Items.Count = 0) Then
            Exit Sub
        End If
        If (ComboBox_EPC6.SelectedItem = Nothing) Then
            Exit Sub
        End If
        str = ComboBox_EPC6.SelectedItem.ToString()
        If (str = "") Then
            Exit Sub
        End If
        ENum1 = Convert.ToByte(str.Length / 4)
        EPCLength = Convert.ToByte(str.Length / 2)
        Dim EPC(ENum1) As Byte
        EPC = HexStringToByteArray(str)
        fPassWord =
HexStringToByteArray(Edit_AccessCode6.Text)
        BlockNum =
Convert.ToByte(ComboBox_BlockNum.SelectedIndex * 2)
        fCmdRet =
StaticClassReaderB.LockUserBlock_G2(fComAdr, EPC, fPassWord,
MaskAdr, MaskLen, MaskFlag, BlockNum, EPCLength, ferrorcode,
frmcomportindex)
        AddCmdLog("LockUserBlock_G2", "Lock User Block",
fCmdRet)
        If (fCmdRet = 0) Then
            StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Lock User Block'Command
Response=0x00" + "(Lock successfully)"
        End If
    End Sub

    Private Sub Form1_FormClosing(ByVal sender As
System.Object, ByVal e As

```

```

System.Windows.Forms.FormClosingEventArgs) Handles
MyBase.FormClosing
    Timer_Test_.Enabled = False
    Timer_G2_Read.Enabled = False
    Timer_G2_Alarm.Enabled = False
    fAppClosed = True
    StaticClassReaderB.CloseComPort()
    If (frmcomportindex > 0) And (radioButton21.Checked) Then
        StaticClassReaderB.CloseNetPort(frmcomportindex)
    End If
End Sub

Private Sub
ComboBox_IntervalTime_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox_IntervalTime.SelectedIndexChanged,
ComboBox_IntervalTime_6B.SelectedIndexChanged
    If (ComboBox_IntervalTime.SelectedIndex < 6) Then
        Timer_Test_.Interval = 100
    Else
        Timer_Test_.Interval =
(ComboBox_IntervalTime.SelectedIndex + 4) * 10
    End If
End Sub

Private Sub SpeedButton_Query_6B_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SpeedButton_Query_6B.Click
    Timer_Test_6B.Enabled = Not Timer_Test_6B.Enabled
    If (Not Timer_Test_6B.Enabled) Then

        If (ListView_ID_6B.Items.Count <> 0) Then
            SpeedButton_Read_6B.Enabled = True
            SpeedButton_Write_6B.Enabled = True
            Button14.Enabled = True
            Button15.Enabled = True
            If (Bycondition_6B.Checked) Then
                Same_6B.Enabled = True
                Different_6B.Enabled = True
            End If
        End If
    End If
End Sub

```

```

        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
If (ListView_ID_6B.Items.Count = 0) Then
    SpeedButton_Read_6B.Enabled = False
    SpeedButton_Write_6B.Enabled = False
    Button14.Enabled = False
    Button15.Enabled = False
    If (Bycondition_6B.Checked) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
AddCmdLog("Inventory", "Exit Query", 0)
If (Byone_6B.Checked) Then
    SpeedButton_Query_6B.Text = "Query "
Else
    SpeedButton_Query_6B.Text = "Query "
End If
Else
    SpeedButton_Read_6B.Enabled = False
    SpeedButton_Write_6B.Enabled = False
    Button14.Enabled = False
    Button15.Enabled = False
    Same_6B.Enabled = False
    Different_6B.Enabled = False
    Less_6B.Enabled = False
    Greater_6B.Enabled = False
    ListView_ID_6B.Items.Clear()
    ComboBox_ID1_6B.Items.Clear()
    CardNum1 = 0
    list.Clear()
    SpeedButton_Query_6B.Text = "Stop"
End If
End Sub

```

```

Public Sub ChangeSubItem1(ByVal ListItem As ListViewItem,
ByVal subItemIndex As Integer, ByVal ItemText As String)
    If (subItemIndex = 1) Then
        If (ListItem.SubItems(subItemIndex).Text <> ItemText)
Then
            ListItem.SubItems(subItemIndex).Text = ItemText
            ListItem.SubItems(subItemIndex + 1).Text = "1"
        Else
            ListItem.SubItems(subItemIndex + 1).Text =
Convert.ToString(Convert.ToInt32(ListItem.SubItems(subItemIndex +
1).Text) + 1)
            If ((Convert.ToUInt32(ListItem.SubItems(subItemIndex
+ 1).Text) > 9999)) Then
                ListItem.SubItems(subItemIndex + 1).Text = "1"
            End If
        End If
    End If
End Sub

```

```

Private Sub Bycondition_6B_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
Bycondition_6B.CheckedChanged
    SpeedButton_Query_6B.Text = "Query"
    If ((Not Timer_6B_Read.Enabled) And (Not
Timer_6B_Write.Enabled) And (Not Timer_Test_6B.Enabled)) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End Sub

```

```

Private Sub Byone_6B_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Byone_6B.CheckedChanged
    SpeedButton_Query_6B.Text = "Query"
    If ((Not Timer_6B_Read.Enabled) And (Not
Timer_6B_Write.Enabled) And (Not Timer_Test_6B.Enabled)) Then
        Same_6B.Enabled = False
    End If
End Sub

```

```

        Different_6B.Enabled = False
        Less_6B.Enabled = False
        Greater_6B.Enabled = False
    End If
End Sub
Private Sub Inventory_6B()
    Dim CardNum As Integer = 0
    Dim ID_6B(2000) As Byte
    Dim ID2_6B(5000) As Byte
    Dim isonlistview As Boolean
    Dim temps As String
    Dim s, ss, sID As String
    Dim aListItem As ListViewItem
    Dim i, j As Integer
    Dim Condition As Byte = 0
    Dim StartAddress As Byte
    Dim mask As Byte = 0
    Dim ConditionContent(300) As Byte
    Dim Contentlen As Byte
    If (Byone_6B.Checked) Then
        fCmdRet = StaticClassReaderB.Inventory_6B(fComAdr,
ID_6B, frmcomportindex)
        If (fCmdRet = 0) Then
            Dim daw(7) As Byte
            Array.Copy(ID_6B, daw, 8)
            temps = ByteArrayToHexString(daw)
            If (list.IndexOf(temps) = -1) Then
                CardNum1 = CardNum1 + 1
                list.Add(temps)
            End If
            While (ListView_ID_6B.Items.Count < CardNum1)

                aListItem =
ListView_ID_6B.Items.Add((ListView_ID_6B.Items.Count
1).ToString())
                aListItem.SubItems.Add("")
                aListItem.SubItems.Add("")
                aListItem.SubItems.Add("")
            End While
        End If
    End If

```



```

        isonlistview = False
        For i = 0 To CardNum1 - 1
            If (temp1 < 0) Then
                ListView_ID_6B.Items(i).SubItems(1).Text = temp1
                aListItem = ListView_ID_6B.Items(i)
                ChangeSubItem1(aListItem, 1, temp1)
                isonlistview = True
            End If
        Next i
        If (Not isonlistview) Then
            aListItem = ListView_ID_6B.Items(CardNum1 - 1)
            s = temp1
            ChangeSubItem1(aListItem, 1, s)
            If (ComboBox_EPC1.Items.IndexOf(s) = -1) Then
                ComboBox_ID1_6B.Items.Add(temp1)
            End If
        End If
        End If

        If (ComboBox_ID1_6B.Items.Count <> 0) Then
            ComboBox_ID1_6B.SelectedIndex = 0
        End If
    End If
    If (Bycondition_6B.Checked) Then
        If (Same_6B.Checked) Then
            Condition = 0
        ElseIf (Different_6B.Checked) Then
            Condition = 1
        ElseIf (Greater_6B.Checked) Then
            Condition = 2
        ElseIf (Less_6B.Checked) Then
            Condition = 3
        End If
        If (Edit_ConditionContent_6B.Text = "") Then
            Exit Sub
        End If
        ss = Edit_ConditionContent_6B.Text
        Contentlen = ss.Length
        Convert.ToByte((Edit_ConditionContent_6B.Text).Length)
    End If
End Sub

```

```

For i = 0 To 15 - Contentlen
    ss = ss + "0"
Next i
Dim Nlen As Integer
Nlen = (ss.Length) / 2
Dim daw(Nlen) As Byte
daw = HexStringToByteArray(ss)
Select Case Contentlen / 2
    Case 1
        mask = &H80
    Case 2
        mask = &HC0
    Case 3
        mask = &HE0
    Case 4
        mask = &HF0
    Case 5
        mask = &HF8
    Case 6
        mask = &HFC
    Case 7
        mask = &HFE
    Case 8
        mask = &HFF
End Select
If (Edit_Query_StartAddress_6B.Text = "") Then
    Exit Sub
End If
StartAddress =
Convert.ToByte(Edit_Query_StartAddress_6B.Text) =
fCmdRet = StaticClassReaderB.inventory2_6B(fComAdr,
Condition, StartAddress, mask, daw, ID2_6B, CardNum,
frmcomportindex)
If ((fCmdRet = &H15) Or (fCmdRet = &H16) Or (fCmdRet
= &H17) Or (fCmdRet = &H18) Or (fCmdRet = &HFB)) Then
    Dim daw1(CardNum * 8) As Byte
    Array.Copy(ID2_6B, daw1, CardNum * 8)
    temps = ByteArrayToHexString(daw1)
    For i = 0 To CardNum - 1

```

```

        sID = temps.Substring(16 * i, 16)
        If ((sID.Length) <> 16) Then
            Exit Sub
        End If
        If (CardNum = 0) Then
            Exit Sub
        End If
        While (ListView_ID_6B.Items.Count < CardNum)
            aListItem =
ListView_ID_6B.Items.Add((ListView_ID_6B.Items.Count +
1).ToString())
            aListItem.SubItems.Add("")
            aListItem.SubItems.Add("")
            aListItem.SubItems.Add("")
        End While
        isonlistview = False
        For j = 0 To ListView_ID_6B.Items.Count - 1
            If (sID =
ListView_ID_6B.Items(j).SubItems(1).Text) Then
                aListItem = ListView_ID_6B.Items(j)
                ChangeSubItem1(aListItem, 1, sID)
                isonlistview = True
            End If
        Next j
        If (Not isonlistview) Then
            aListItem = ListView_ID_6B.Items(i)
            s = sID
            ChangeSubItem1(aListItem, 1, s)
            If (ComboBox_EPC1.Items.IndexOf(s) = -1) Then
                ComboBox_ID1_6B.Items.Add(sID)
            End If
        End If
    Next i
    If (ComboBox_ID1_6B.Items.Count <> 0) Then
        ComboBox_ID1_6B.SelectedIndex = 0
    End If
End If
If (Timer_Test_6B.Enabled) Then

```

```

        If (Bycondition_6B.Checked) Then
            If (fCmdRet <> 0) Then
                AddCmdLog("Inventory", "Query tag", fCmdRet)
            ElseIf (fCmdRet = &HFB) Then
                StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Query Tag'Command
Response=0xFB" + "(No Tag Operable)"
            ElseIf (fCmdRet = 0) Then
                StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Query Tag'Command
Response=0x00" + "(Find a Tag)"
            Else
                AddCmdLog("Inventory", "Query Tag", fCmdRet)
            End If
            If (fCmdRet = &HEE) Then
                StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Query Tag'Command
Response=0xee" + "(Response Command Error)"
            End If
        End If
    End If
    If (fAppClosed) Then
        Close()
    End If
End Sub

Private Sub Timer_Test_6B_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer_Test_6B.Tick
    If (fisinventoryscan_6B) Then
        Exit Sub
    End If
    fisinventoryscan_6B = True
    Inventory_6B()
    fisinventoryscan_6B = False
End Sub

```

```

Private Sub SpeedButton_Read_6B_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SpeedButton_Read_6B.Click
    If ((Edit_StartAddress_6B.Text = "") Or (Edit_Len_6B.Text =
"")) Then
        MessageBox.Show("Start address or length is empty!Please
input!", "Information")
        Exit Sub
    End If
    Timer_6B_Read.Enabled = Not Timer_6B_Read.Enabled
    If (Not Timer_6B_Read.Enabled) Then
        AddCmdLog("Read", "Exit Read", 0)
        SpeedButton_Read_6B.Text = "Read "
        SpeedButton_Query_6B.Enabled = True
        SpeedButton_Write_6B.Enabled = True
        Button14.Enabled = True
        Button15.Enabled = True
        If (Bycondition_6B.Checked) Then
            Same_6B.Enabled = True
            Different_6B.Enabled = True
            Less_6B.Enabled = True
            Greater_6B.Enabled = True
        End If
    Else
        SpeedButton_Query_6B.Enabled = False
        SpeedButton_Write_6B.Enabled = False
        Button14.Enabled = False
        Button15.Enabled = False
        If (Bycondition_6B.Checked) Then
            Same_6B.Enabled = False
            Different_6B.Enabled = False
            Less_6B.Enabled = False
            Greater_6B.Enabled = False
        End If
        SpeedButton_Read_6B.Text = "Stop"
    End If
End Sub
Private Sub Read_6B()
    Dim temp, temps As String

```

```

Dim CardData(320) As Byte
Dim ID_6B(7) As Byte
Dim Num, StartAddress As Byte
If (ComboBox_ID1_6B.Items.Count = 0) Then
    Exit Sub
End If
If (ComboBox_ID1_6B.SelectedItem = Nothing) Then
    Exit Sub
End If
temp = ComboBox_ID1_6B.SelectedItem.ToString()
If (temp = "") Then
    Exit Sub
End If
ID_6B = HexStringToByteArray(temp)
If (Edit_StartAddress_6B.Text = "") Then
    Exit Sub
End If
StartAddress = Convert.ToByte(Edit_StartAddress_6B.Text,
16)
If (Edit_Len_6B.Text = "") Then
    Exit Sub
End If
Num = Convert.ToByte(Edit_Len_6B.Text)
fCmdRet = StaticClassReaderB.ReadCard_6B(fComAdr,
ID_6B, StartAddress, Num, CardData, ferrortcode, frmcomportindex)
If (fCmdRet = 0) Then
    Dim data(Num - 1) As Byte
    Array.Copy(CardData, data, Num)
    temps = ByteArrayToHexString(data)
    listBox2.Items.Add(temps)
End If
If (fAppClosed) Then
    Close()
End If
End Sub

Private Sub Timer_6B_Read_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer_6B_Read.Tick

```

```

    If (fTimer_6B_ReadWrite) Then
        Exit Sub
    End If
    fTimer_6B_ReadWrite = True
    Read_6B()
    fTimer_6B_ReadWrite = False
End Sub

Private Sub SpeedButton_Write_6B_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SpeedButton_Write_6B.Click
    If ((Edit_WriteData_6B.Text = "") Or
((Edit_WriteData_6B.Text.Length Mod 2) <> 0)) Then
        MessageBox.Show("Please input in bytes in hexadecimal
form!", "Information")
        Exit Sub
    End If
    If ((Edit_StartAddress_6B.Text = "") Or (Edit_Len_6B.Text =
"")) Then
        MessageBox.Show("Start address or length is empty!Please
input!", "Information")
        Exit Sub
    End If
    Timer_6B_Write.Enabled = Not Timer_6B_Write.Enabled
    If (Not Timer_6B_Write.Enabled) Then
        AddCmdLog("Write", "Exit Write", 0)
        SpeedButton_Write_6B.Text = "Write "
        SpeedButton_Query_6B.Enabled = True
        SpeedButton_Read_6B.Enabled = True
        Button14.Enabled = True
        Button15.Enabled = True
        If (Bycondition_6B.Checked) Then
            Same_6B.Enabled = True
            Different_6B.Enabled = True
            Less_6B.Enabled = True
            Greater_6B.Enabled = True
        End If
    Else
        SpeedButton_Query_6B.Enabled = False
    End If
End Sub

```

```

SpeedButton_Read_6B.Enabled = False
Button14.Enabled = False
Button15.Enabled = False
If (Bycondition_6B.Checked) Then
    Same_6B.Enabled = False
    Different_6B.Enabled = False
    Less_6B.Enabled = False
    Greater_6B.Enabled = False
End If
SpeedButton_Write_6B.Text = "Stop"
End If
End Sub
Private Sub Write_6B()
    Dim temp As String
    Dim CardData(320) As Byte
    Dim ID_6B(7) As Byte
    Dim StartAddress As Byte
    Dim Writedatalen As Byte
    Dim writtenbyte As Integer = 0
    If (ComboBox_ID1_6B.Items.Count = 0) Then
        Exit Sub
    End If
    If (ComboBox_ID1_6B.SelectedItem = Nothing) Then
        Exit Sub
    End If
    temp = ComboBox_ID1_6B.SelectedItem.ToString()
    If (temp = "") Then
        Exit Sub
    End If
    ID_6B = HexStringToByteArray(temp)
    If (Edit_StartAddress_6B.Text = "") Then
        Exit Sub
    End If
    StartAddress = Convert.ToByte(Edit_StartAddress_6B.Text,
16)
    If ((Edit_WriteData_6B.Text = "") Or
(Edit_WriteData_6B.Text.Length Mod 2) <> 0) Then
        Exit Sub
    End If

```



```

        Writedatalen =
Convert.ToByte(Edit_WriteData_6B.Text.Length)
        Dim Writedata(Writedatalen) As Byte
        Writedata =
HexStringToByteArray(Edit_WriteData_6B.Text)
        fCmdRet = StaticClassReaderB.WriteCard_6B(fComAdr,
ID_6B, StartAddress, Writedata, Writedatalen, writtenbyte, ferrortcode,
frmcomportindex)
        AddCmdLog("WriteCard", "Stop", fCmdRet)
        If (fAppClosed) Then
            Close()
        End If
    End Sub

    Private Sub Timer_6B_Write_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Timer_6B_Write.Tick
        If (fTimer_6B_ReadWrite) Then
            Exit Sub
        End If
        fTimer_6B_ReadWrite = True
        Write_6B()
        fTimer_6B_ReadWrite = False
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button14.Click
        Dim Address As Byte
        Dim temps As String
        Dim ID_6B(7) As Byte
        If (ComboBox_ID1_6B.Items.Count = 0) Then
            Exit Sub
        End If
        If (ComboBox_ID1_6B.SelectedItem = Nothing) Then
            Exit Sub
        End If
        temps = ComboBox_ID1_6B.SelectedItem.ToString()
        If (temps = "") Then
            Exit Sub
        End If
    End Sub

```

```

End If
ID_6B = HexStringToByteArray(temps)
If (Edit_StartAddress_6B.Text = "") Then
    Exit Sub
End If
Address = Convert.ToByte(Edit_StartAddress_6B.Text)
If (MessageBox.Show(Me, "permanently Lock the address
Confirmed?", "Information", MessageBoxButtons.OKCancel) =
DialogResult.Cancel) Then
    Exit Sub
End If
fCmdRet = StaticClassReaderB.LockByte_6B(fComAdr,
ID_6B, Address, ferrortcode, frmcomportindex)
AddCmdLog("LockByte_6B", "Lock", fCmdRet)
End Sub

```

```

Private Sub Button15_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button15.Click
    Dim Address As Byte
    Dim ReLockState As Byte = 2
    Dim temps As String
    Dim ID_6B(7) As Byte
    If (ComboBox_ID1_6B.Items.Count = 0) Then
        Exit Sub
    End If
    If (ComboBox_ID1_6B.SelectedItem = Nothing) Then
        Exit Sub
    End If
    temps = ComboBox_ID1_6B.SelectedItem.ToString()
    If (temps = "") Then
        Exit Sub
    End If
    ID_6B = HexStringToByteArray(temps)
    If (Edit_StartAddress_6B.Text = "") Then
        Exit Sub
    End If
    Address = Convert.ToByte(Edit_StartAddress_6B.Text)
    fCmdRet = StaticClassReaderB.CheckLock_6B(fComAdr,
ID_6B, Address, ReLockState, ferrortcode, frmcomportindex)

```

```

AddCmdLog("CheckLock_6B", "Check Lock", fCmdRet)
If (fCmdRet = 0) Then
    If (ReLockState = 0) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Check Lock'Command
Response=0x00" + "(The Byte is unlocked)"
    End If
    If (ReLockState = 1) Then
        StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + " 'Check Lock'Command
Response=0x01" + "(The Byte is locked)"
    End If
End If
End Sub

```

```

Private Sub Button22_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button22.Click
    listBox2.Items.Clear()
End Sub

```

```

Private Sub C_Reserve_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
C_Reserve.CheckedChanged
    If ((Not Timer_Test_.Enabled) And (Not
Timer_G2_Alarm.Enabled) And (Not Timer_G2_Read.Enabled)) Then
        If (ListView1_EPC.Items.Count <> 0) Then
            Button_DataWrite.Enabled = True
        End If
    End If
    Edit_WordPtr.ReadOnly = False
End Sub

```

```

Private Sub C_TID_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
C_TID.CheckedChanged
    If ((Not Timer_Test_.Enabled) And (Not
Timer_G2_Alarm.Enabled) And (Not Timer_G2_Read.Enabled)) Then
        If (ListView1_EPC.Items.Count <> 0) Then
            Button_DataWrite.Enabled = True
        End If
    End If
End Sub

```

```

        End If
    End If
    Edit_WordPtr.ReadOnly = False
End Sub

```

```

Private Sub C_User_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
C_User.CheckedChanged
    If ((Not Timer_Test.Enabled) And (Not
Timer_G2_Alarm.Enabled) And (Not Timer_G2_Read.Enabled)) Then
        If (ListView1_EPC.Items.Count <> 0) Then
            Button_DataWrite.Enabled = True
        End If
    End If
    Edit_WordPtr.ReadOnly = False
End Sub

```

```

Private Sub C_EPC_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
C_EPC.CheckedChanged
    If ((Not Timer_Test.Enabled) And (Not
Timer_G2_Alarm.Enabled) And (Not Timer_G2_Read.Enabled)) Then
        If (ListView1_EPC.Items.Count <> 0) Then
            Button_DataWrite.Enabled = True
        End If
    End If
    If (checkBox_pc.Checked) Then
        Edit_WordPtr.Text = "02"
        Edit_WordPtr.ReadOnly = True
    Else
        Edit_WordPtr.ReadOnly = False
    End If
End Sub

```

```

Private Sub tabControl1_Selecting(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.TabControlCancelEventArgs)
    timer1.Enabled = False
    button10.Text = "Get"

```

```

Timer_G2_Alarm.Enabled = False
Timer_G2_Read.Enabled = False
Timer_Test_.Enabled = False
SpeedButton_Read_G2.Text = "Read"
button2.Text = "Query Tag"
button4.Text = "Check Alarm"
If ((ListView1_EPC.Items.Count <> 0) And (ComOpen))

```

Then

```

    button2.Enabled = True
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = True
    Proect2.Enabled = True
    Always2.Enabled = True
    AlwaysNot2.Enabled = True
    P_Reserve.Enabled = True
    P_EPC.Enabled = True
    P_TID.Enabled = True
    P_User.Enabled = True
    Button_DestroyCard.Enabled = True
    Button_SetReadProtect_G2.Enabled = True
    Button_SetEASAlarm_G2.Enabled = True
    Alarm_G2.Enabled = True
    NoAlarm_G2.Enabled = True
    Button_LockUserBlock_G2.Enabled = True
    Button_WriteEPC_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True
    button4.Enabled = True
    SpeedButton_Read_G2.Enabled = True
    Button_SetProtectState.Enabled = True
    Button_DataWrite.Enabled = True
    Button_BlockWrite.Enabled = True

```

```

    Button_BlockErase.Enabled = True
End If
If ((ListView1_EPC.Items.Count = 0) And (ComOpen)) Then
    button2.Enabled = True
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = False
    Proect2.Enabled = False
    Always2.Enabled = False
    AlwaysNot2.Enabled = False
    P_Reserve.Enabled = False
    P_EPC.Enabled = False
    P_TID.Enabled = False
    P_User.Enabled = False
    Button_DestroyCard.Enabled = False
    Button_SetReadProtect_G2.Enabled = False
    Button_SetEASAlarm_G2.Enabled = False
    Alarm_G2.Enabled = False
    NoAlarm_G2.Enabled = False
    Button_LockUserBlock_G2.Enabled = False
    SpeedButton_Read_G2.Enabled = False
    Button_DataWrite.Enabled = False
    Button_BlockErase.Enabled = False
    Button_BlockWrite.Enabled = False
    Button_WriteEPC_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True
    button4.Enabled = True
    Button_SetProtectState.Enabled = False
End If

Timer_Test_6B.Enabled = False
Timer_6B_Read.Enabled = False
Timer_6B_Write.Enabled = False

```

```

SpeedButton_Query_6B.Text = "Query"
SpeedButton_Read_6B.Text = "Read"
SpeedButton_Write_6B.Text = "Write"
If ((ListView_ID_6B.Items.Count <> 0) And (ComOpen))
Then
    SpeedButton_Query_6B.Enabled = True
    SpeedButton_Read_6B.Enabled = True
    SpeedButton_Write_6B.Enabled = True
    Button14.Enabled = True
    Button15.Enabled = True
    If (Bycondition_6B.Checked) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
If ((ListView_ID_6B.Items.Count = 0) And (ComOpen)) Then
    SpeedButton_Query_6B.Enabled = True
    SpeedButton_Read_6B.Enabled = False
    SpeedButton_Write_6B.Enabled = False
    Button14.Enabled = False
    Button15.Enabled = False
    If (Bycondition_6B.Checked) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
End Sub

```

```

Private Sub Edit_CmdComAddr_KeyPress(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles
Edit_WriteData.KeyPress, Edit_WordPtr.KeyPress,
Edit_AccessCode2.KeyPress, textBox2.KeyPress,
Edit_AccessCode6.KeyPress, Edit_AccessCode5.KeyPress,
Edit_AccessCode4.KeyPress, Edit_AccessCode3.KeyPress,

```

```

Edit_WriteEPC.KeyPress,           Edit_DestroyCode.KeyPress,
Edit_WriteData_6B.KeyPress,      Edit_StartAddress_6B.KeyPress,
maskLen_textBox.KeyPress,       maskadr_textbox.KeyPress,
textBox5.KeyPress, textBox4.KeyPress
    e.Handled =
("0123456789ABCDEF_".IndexOf(Char.ToUpper(e.KeyChar)) < 0)
End Sub

```

```

Private Sub Edit_Len_6B_KeyPress(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles
Edit_Len_6B.KeyPress, textBox1.KeyPress
    e.Handled =
("0123456789_".IndexOf(Char.ToUpper(e.KeyChar)) < 0)
End Sub

```

```

Private Sub comboBox4_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
comboBox4.SelectedIndexChanged

```

```

    Dim i = 0

```

```

    If (comboBox4.SelectedIndex = 0) Then

```

```

        radioButton5.Enabled = False
        radioButton6.Enabled = False
        radioButton7.Enabled = False
        radioButton8.Enabled = False
        radioButton9.Enabled = False
        radioButton10.Enabled = False
        radioButton11.Enabled = False
        radioButton12.Enabled = False
        radioButton13.Enabled = False
        radioButton14.Enabled = False
        radioButton15.Enabled = False
        RadioButton16.Enabled = False
        RadioButton17.Enabled = False
        RadioButton18.Enabled = False
        RadioButton19.Enabled = False
        radioButton20.Enabled = False
        textBox3.Enabled = False
        comboBox5.Enabled = False
    End If
End Sub

```



```

        ComboBox6.Enabled = False
    End If
    If ((comboBox4.SelectedIndex = 1) Or
(comboBox4.SelectedIndex = 2) Or (comboBox4.SelectedIndex = 3))
Then
        radioButton5.Enabled = True
        radioButton6.Enabled = True
        radioButton7.Enabled = True
        radioButton8.Enabled = True
        radioButton20.Enabled = True
        comboBox5.Items.Clear()
    If (radioButton20.Checked) Then
        For i = 1 To 4
            comboBox5.Items.Add(Convert.ToString(i))
        Next i
        comboBox5.SelectedIndex = 3
        label42.Text = "Read Byte Number:"
    Else
        For i = 1 To 32
            comboBox5.Items.Add(Convert.ToString(i))
        Next i
        comboBox5.SelectedIndex = 0
        label42.Text = "Read Word Number:"
    End If

    If (radioButton7.Checked) Then
        RadioButton16.Enabled = True
        RadioButton17.Enabled = True
    Else
        RadioButton16.Enabled = False
        RadioButton17.Enabled = False
    End If
    If (radioButton5.Checked) Then
        radioButton9.Enabled = True
        radioButton10.Enabled = True
        radioButton11.Enabled = True
        radioButton12.Enabled = True
        RadioButton18.Enabled = True
    If (radioButton20.Checked) Then

```

```

        radioButton13.Enabled = False
        RadioButton19.Enabled = False
    Else
        radioButton13.Enabled = True
        RadioButton19.Enabled = True
    End If
    If ((radioButton13.Checked) Or
(RadioButton19.Checked)) Then
        ComboBox6.Enabled = False
    Else
        ComboBox6.Enabled = True
    End If
Else
    ComboBox6.Enabled = True
End If
radioButton14.Enabled = True
radioButton15.Enabled = True
textBox3.Enabled = True
If (radioButton7.Checked) Then
    comboBox5.Enabled = False
Else
    comboBox5.Enabled = True
End If

End If
End Sub

Private Sub radioButton5_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radioButton6.CheckedChanged, radioButton5.CheckedChanged
    If (radioButton5.Checked) Then
        If ((comboBox4.SelectedIndex = 1) Or
(comboBox4.SelectedIndex = 2) Or (comboBox4.SelectedIndex = 3))
Then
            radioButton9.Enabled = True
            radioButton10.Enabled = True
            radioButton11.Enabled = True
            radioButton12.Enabled = True

```

```

radioButton13.Enabled = True
RadioButton18.Enabled = True
If (RadioButton16.Checked) Then
    label41.Text = "First Word Addr(Hex):"
Else
    label41.Text = "First Byte Addr(Hex):"
End If
If (radioButton20.Checked) Then
    radioButton13.Enabled = False
    RadioButton19.Enabled = False
    label41.Text = "First Byte Addr(Hex):"
Else
    radioButton13.Enabled = True
    RadioButton19.Enabled = True
End If
If (radioButton7.Checked) Then
    RadioButton16.Enabled = True
    RadioButton17.Enabled = True
    If ((radioButton13.Checked) Or
(RadioButton19.Checked)) Then
        ComboBox6.Enabled = False
    Else
        ComboBox6.Enabled = True
    End If
Else
    RadioButton16.Enabled = False
    RadioButton17.Enabled = False
    If ((radioButton13.Checked) Or
(RadioButton19.Checked)) Then
        ComboBox6.Enabled = False
    Else
        ComboBox6.Enabled = True
    End If
    If (radioButton20.Checked) Then
        label41.Text = "First Byte Addr(Hex):"
    Else
        label41.Text = "First Word Addr(Hex):"
    End If
End If

```

```

End If
Else
    radioButton9.Enabled = False
    radioButton10.Enabled = False
    radioButton11.Enabled = False
    radioButton12.Enabled = False
    radioButton13.Enabled = False
    RadioButton18.Enabled = False
    RadioButton16.Enabled = False
    RadioButton17.Enabled = False
    RadioButton19.Enabled = False
    ComboBox6.Enabled = True
    label41.Text = "First Byte Addr(Hex)"
End If
End Sub

```

```

Private Sub radioButton7_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radioButton7.CheckedChanged
    If (radioButton5.Checked) And (comboBox4.SelectedIndex >
0) Then
        RadioButton16.Enabled = True
        RadioButton17.Enabled = True
        radioButton13.Enabled = True
        RadioButton19.Enabled = True
        If (RadioButton16.Checked) Then
            label41.Text = "First Word Addr(Hex):"
        Else
            label41.Text = "First Byte Addr(Hex):"
        End If
        label42.Text = "Read Word Number:"
    End If
    comboBox5.Enabled = False
End Sub

```

```

Private Sub radioButton8_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radioButton8.CheckedChanged, radioButton20.CheckedChanged
    Dim i = 0

```

```

    If ((comboBox4.SelectedIndex = 1) Or
(comboBox4.SelectedIndex = 2) Or (comboBox4.SelectedIndex = 3))
Then
    If (radioButton8.Checked) Then
        comboBox5.Enabled = True
    End If
    comboBox5.Items.Clear()
    If (radioButton20.Checked) Then
        For i = 1 To 4
            comboBox5.Items.Add(Convert.ToString(i))
        Next i
        comboBox5.SelectedIndex = 3
        label42.Text = "Read Byte Number:"
        comboBox5.Enabled = True
        label41.Text = "First Byte Addr(Hex):"
    Else
        For i = 1 To 32
            comboBox5.Items.Add(Convert.ToString(i))
        Next i
        comboBox5.SelectedIndex = 0
        label42.Text = "Read Word Number:"
        label41.Text = "First Word Addr(Hex):"
    End If
    If (radioButton5.Checked) Then
        RadioButton16.Enabled = False
        RadioButton17.Enabled = False
        If (radioButton20.Checked) Then
            radioButton13.Enabled = False
            RadioButton19.Enabled = False
        Else
            radioButton13.Enabled = True
            RadioButton19.Enabled = True
        End If
    Else
        label41.Text = "First Byte Addr(Hex):"
        radioButton13.Enabled = False
        RadioButton19.Enabled = False
    End If
End If

```

End Sub

```
Private Sub button6_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button6.Click
    Dim Wg_mode As Byte = 0
    Dim Wg_Data_Interval As Byte
    Dim Wg_Pulse_Width As Byte
    Dim Wg_Pulse_Interval As Byte
    If (radioButton1.Checked) Then
        If (radioButton3.Checked) Then
            Wg_mode = 2
        Else
            Wg_mode = 0
        End If
    End If
    If (radioButton2.Checked) Then
        If (radioButton3.Checked) Then
            Wg_mode = 3
        Else
            Wg_mode = 1
        End If
    End If
    Wg_Data_Interval =
Convert.ToByte(comboBox1.SelectedIndex)
    Wg_Pulse_Width =
Convert.ToByte(comboBox3.SelectedIndex + 1)
    Wg_Pulse_Interval =
Convert.ToByte(comboBox2.SelectedIndex + 1)
    fCmdRet = StaticClassReaderB.SetWGParameter(fComAdr,
Wg_mode, Wg_Data_Interval, Wg_Pulse_Width, Wg_Pulse_Interval,
frmcomportindex)
    AddCmdLog("SetWGParameter", "SetWGParameter",
fCmdRet)
End Sub
```

```
Private Sub button8_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button8.Click
    Dim Reader_bit0, Reader_bit1, Reader_bit2, Reader_bit3,
Reader_bit4 As Integer
```

```

Dim Parameter(5) As Byte
Parameter(0) = Convert.ToByte(comboBox4.SelectedIndex)
If (radioButton5.Checked) Then
    Reader_bit0 = 0
Else
    Reader_bit0 = 1
End If
If (radioButton7.Checked) Then
    Reader_bit1 = 0
Else
    Reader_bit1 = 1
End If
If (radioButton14.Checked) Then
    Reader_bit2 = 0
Else
    Reader_bit2 = 1
End If
If (RadioButton16.Checked) Then
    Reader_bit3 = 0
Else
    Reader_bit3 = 1
End If
If (radioButton20.Checked) Then
    Reader_bit4 = 1
Else
    Reader_bit4 = 0
End If
Parameter(1) = Convert.ToByte(Reader_bit0 * 1 +
Reader_bit1 * 2 + Reader_bit2 * 4 + Reader_bit3 * 8 + Reader_bit4 * 16)
If (radioButton9.Checked) Then
    Parameter(2) = 0
End If
If (radioButton10.Checked) Then
    Parameter(2) = 1
End If
If (radioButton11.Checked) Then
    Parameter(2) = 2
End If
If (radioButton12.Checked) Then

```

```

        Parameter(2) = 3
    End If
    If (radioButton13.Checked) Then
        Parameter(2) = 4
    End If
    If (RadioButton18.Checked) Then
        Parameter(2) = 5
    End If
    If (RadioButton19.Checked) Then
        Parameter(2) = 6
    End If
    If (textBox3.Text = "") Then
        MessageBox.Show("Address is NULL!", "Information")
        Exit Sub
    End If
    Parameter(3) = Convert.ToByte(textBox3.Text, 16)
    Parameter(4) = Convert.ToByte(comboBox5.SelectedIndex +
1)
    Parameter(5) = Convert.ToByte(ComboBox6.SelectedIndex)
    fCmdRet = StaticClassReaderB.SetWorkMode(fComAdr,
Parameter, frmcomportindex)
    If (fCmdRet = 0) Then
        If ((comboBox4.SelectedIndex = 1) Or
(comboBox4.SelectedIndex = 2) Or (comboBox4.SelectedIndex = 3))
Then
            If (radioButton6.Checked) Then
                radioButton13.Enabled = False
                RadioButton19.Enabled = False
            Else
                If (radioButton20.Checked) Then
                    radioButton13.Enabled = False
                    RadioButton19.Enabled = False
                End If
            End If
            button10.Enabled = True
            button11.Enabled = True
        End If
        If (comboBox4.SelectedIndex = 0) Then
            button10.Enabled = False

```



```

        button11.Enabled = False
        button10.Text = "Get"
        timer1.Enabled = False
    End If
End If
AddCmdLog("SetWorkMode", "SetWorkMode", fCmdRet)
End Sub

Private Sub button10_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button10.Click
    timer1.Enabled = Not timer1.Enabled
    If (Not timer1.Enabled) Then
        button10.Text = "Get"
    Else
        button10.Text = "Stop"
    End If
End Sub
Private Sub GetData()
    Dim ScanModeData(40960) As Byte
    Dim ValidDataLength, i As Integer
    Dim temp, temps As String
    ValidDataLength = 0
    fCmdRet
StaticClassReaderB.ReadActiveModeData(ScanModeData,
ValidDataLength, frmcomportindex)
    If (fCmdRet = 0) Then
        temp = ""
        temps = ByteArrayToHexString(ScanModeData)
        For i = 0 To ValidDataLength - 1
            temp = temp + temps.Substring(i * 2, 2) + " "
        Next i
        listBox3.Items.Add(temp)
        listBox3.SelectedIndex = listBox3.Items.Count - 1
    End If
    AddCmdLog("Get", "Get", fCmdRet)
End Sub

```

```

Private Sub timer1_Tick(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles timer1.Tick
    If (flsInventoryScan) Then
        flsInventoryScan = True
    End If
    GetData()
    If (fAppClosed) Then
        Close()
    End If
    flsInventoryScan = False
End Sub

```

```

Private Sub button11_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button11.Click
    listBox3.Items.Clear()
End Sub

```

```

Private Sub radioButton_band1_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
radioButton_band1.CheckedChanged
    Dim i As Integer
    ComboBox_dmaxfre.Items.Clear()
    ComboBox_dminfre.Items.Clear()
    For i = 0 To 62
        ComboBox_dminfre.Items.Add(Convert.ToString(902.6 + i
* 0.4) + " MHz")
        ComboBox_dmaxfre.Items.Add(Convert.ToString(902.6 +
i * 0.4) + " MHz")
    Next i
    ComboBox_dmaxfre.SelectedIndex = 62
    ComboBox_dminfre.SelectedIndex = 0
End Sub

```

```

Private Sub radioButton_band2_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
radioButton_band2.CheckedChanged
    Dim i As Integer
    ComboBox_dmaxfre.Items.Clear()
    ComboBox_dminfre.Items.Clear()

```

```

        For i = 0 To 19
            ComboBox_dminfre.Items.Add(Convert.ToString(920.125
+ i * 0.25) + " MHz")
            ComboBox_dmaxfre.Items.Add(Convert.ToString(920.125
+ i * 0.25) + " MHz")
        Next i
        ComboBox_dmaxfre.SelectedIndex = 19
        ComboBox_dminfre.SelectedIndex = 0
    End Sub

```

```

    Private Sub radioButton_band3_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
radioButton_band3.CheckedChanged
        Dim i As Integer
        ComboBox_dmaxfre.Items.Clear()
        ComboBox_dminfre.Items.Clear()
        For i = 0 To 49
            ComboBox_dminfre.Items.Add(Convert.ToString(902.75 +
i * 0.5) + " MHz")
            ComboBox_dmaxfre.Items.Add(Convert.ToString(902.75
+ i * 0.5) + " MHz")
        Next i
        ComboBox_dmaxfre.SelectedIndex = 49
        ComboBox_dminfre.SelectedIndex = 0
    End Sub

```

```

    Private Sub radioButton_band4_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
radioButton_band4.CheckedChanged
        Dim i As Integer
        ComboBox_dmaxfre.Items.Clear()
        ComboBox_dminfre.Items.Clear()
        For i = 0 To 31
            ComboBox_dminfre.Items.Add(Convert.ToString(917.1 + i
* 0.2) + " MHz")
            ComboBox_dmaxfre.Items.Add(Convert.ToString(917.1 +
i * 0.2) + " MHz")
        Next i
        ComboBox_dmaxfre.SelectedIndex = 31
    End Sub

```

```
        ComboBox_dminfre.SelectedIndex = 0
    End Sub
```

```
    Private Sub checkBox1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
checkBox1.CheckedChanged
        If (checkBox1.Checked) Then
            maskadr_textbox.Enabled = True
            maskLen_textBox.Enabled = True
        Else
            maskadr_textbox.Enabled = False
            maskLen_textBox.Enabled = False
        End If
    End Sub
```

```
    Private Sub radioButton9_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radioButton9.CheckedChanged, radioButton10.CheckedChanged,
RadioButton18.CheckedChanged, radioButton12.CheckedChanged,
radioButton11.CheckedChanged
        ComboBox6.Enabled = True
    End Sub
```

```
    Private Sub radioButton13_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
radioButton13.CheckedChanged, RadioButton19.CheckedChanged
        ComboBox6.Enabled = False
    End Sub
```

```
    Private Sub RadioButton16_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton16.CheckedChanged
        label41.Text = "First Word Addr(Hex):"
    End Sub
```

```
    Private Sub RadioButton17_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton17.CheckedChanged
        label41.Text = "First Byte Addr(Hex):"
```

End Sub

```
Private Sub button9_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button9.Click
    Dim Parameter(20) As Byte
    fCmdRet =
StaticClassReaderB.GetWorkModeParameter(fComAdr, Parameter,
frmcomportindex)
    If (fCmdRet = 0) Then
        If (Parameter(0) = 0) Then
            radioButton1.Checked = True
            radioButton4.Checked = True
        End If
        If (Parameter(0) = 1) Then
            radioButton2.Checked = True
            radioButton4.Checked = True
        End If
        If (Parameter(0) = 2) Then
            radioButton1.Checked = True
            radioButton3.Checked = True
        End If
        If (Parameter(0) = 3) Then
            radioButton2.Checked = True
            radioButton3.Checked = True
        End If
        comboBox1.SelectedIndex =
Convert.ToInt32(Parameter(1))
        comboBox2.SelectedIndex =
Convert.ToInt32(Parameter(3) - 1)
        comboBox3.SelectedIndex =
Convert.ToInt32(Parameter(2) - 1)
        comboBox4.SelectedIndex =
Convert.ToInt32(Parameter(4))
        If ((Parameter(4) = 1) Or (Parameter(4) = 2) Or
(Parameter(4) = 3)) Then
            button10.Enabled = True
            button11.Enabled = True
            radioButton5.Enabled = True
```

```

radioButton6.Enabled = True
radioButton7.Enabled = True
radioButton8.Enabled = True
If (radioButton5.Checked) Then
    If (radioButton7.Checked) Then
        RadioButton16.Enabled = True
        RadioButton17.Enabled = True
    Else
        RadioButton16.Enabled = False
        RadioButton17.Enabled = False
    End If
radioButton9.Enabled = True
radioButton10.Enabled = True
radioButton11.Enabled = True
radioButton12.Enabled = True
RadioButton18.Enabled = True
radioButton20.Enabled = True
If (Convert.ToInt32((Parameter(5) And &H10)) =
&H10) Then
    radioButton13.Enabled = False
    RadioButton19.Enabled = False
    Else
        radioButton13.Enabled = True
        RadioButton19.Enabled = True
    End If
    If ((radioButton13.Checked) Or
(RadioButton19.Checked)) Then
        ComboBox6.Enabled = False
    Else
        ComboBox6.Enabled = True
    End If
Else
    ComboBox6.Enabled = True
    radioButton14.Enabled = True
    radioButton15.Enabled = True
    textBox3.Enabled = True
End If
If ((radioButton8.Checked) Or (radioButton20.Checked))
Then

```

```

        comboBox5.Enabled = True
    End If
    If (Parameter(4) = 0) Then
        button10.Enabled = False
        button11.Enabled = False
        radioButton5.Enabled = False
        radioButton6.Enabled = False
        radioButton7.Enabled = False
        radioButton8.Enabled = False
        radioButton9.Enabled = False
        radioButton10.Enabled = False
        radioButton11.Enabled = False
        radioButton12.Enabled = False
        radioButton13.Enabled = False
        radioButton14.Enabled = False
        radioButton15.Enabled = False
        RadioButton16.Enabled = False
        RadioButton17.Enabled = False
        RadioButton18.Enabled = False
        RadioButton19.Enabled = False
        radioButton20.Enabled = False
        textBox3.Enabled = False
        comboBox5.Enabled = False
        ComboBox6.Enabled = False
    End If
End If
If (Convert.ToInt32((Parameter(5)) And &H1) = 0) Then
    radioButton5.Checked = True
Else
    radioButton6.Checked = True
End If
If (Convert.ToInt32((Parameter(5)) And &H2) = 0) Then
    radioButton7.Checked = True
Else
    If (Convert.ToInt32((Parameter(5)) And &H10) = 0)
Then
        radioButton8.Checked = True
    Else
        radioButton20.Checked = True
    End If
End If

```

```

        End If
    End If
    If (Convert.ToInt32((Parameter(5)) And &H4) = 0) Then
        radioButton14.Checked = True
    Else
        radioButton15.Checked = True
    End If
    If (Convert.ToInt32((Parameter(5)) And &H8) = 0) Then
        RadioButton16.Checked = True
    Else
        RadioButton17.Checked = True
    End If
    Select Case (Parameter(6))
        Case 0
            radioButton9.Checked = True
        Case 1
            radioButton10.Checked = True
        Case 2
            radioButton11.Checked = True
        Case 3
            radioButton12.Checked = True
        Case 4
            radioButton13.Checked = True
        Case 5
            RadioButton18.Checked = True
        Case 6
            RadioButton19.Checked = True
    End Select
    textBox3.Text      =      Convert.ToString(Parameter(7),
16).PadLeft(2, "0")
    comboBox5.SelectedIndex      =
Convert.ToInt32(Parameter(8) - 1)
    ComboBox6.SelectedIndex      =
Convert.ToInt32(Parameter(9))
    ComboBox7.SelectedIndex      =
Convert.ToInt32(Parameter(10))
    comboBox_OffsetTime.SelectedIndex      =
Convert.ToInt32(Parameter(11))
    End If

```



```

        AddCmdLog("SetAccuracy", "GetWorkModeparameter",
fCmdRet)
    End Sub

```

```

    Private Sub Button12_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button12.Click
        Dim Accuracy As Byte
        Accuracy = Convert.ToByte(ComboBox7.SelectedIndex)
        fCmdRet = StaticClassReaderB.SetAccuracy(fComAdr,
Accuracy, frmcomportindex)
        AddCmdLog("SetAccuracy", "SetAccuracy", fCmdRet)
    End Sub

```

```

    Private Sub ComboBox_COM_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
ComboBox_AlreadyOpenCOM.SelectedIndexChanged
        ComboBox_baud2.Items.Clear()
        If (ComboBox_COM.SelectedIndex = 0) Then
            ComboBox_baud2.Items.Add("9600bps")
            ComboBox_baud2.Items.Add("19200bps")
            ComboBox_baud2.Items.Add("38400bps")
            ComboBox_baud2.Items.Add("57600bps")
            ComboBox_baud2.Items.Add("115200bps")
            ComboBox_baud2.SelectedIndex = 3
        Else
            ComboBox_baud2.Items.Add("Auto")
            ComboBox_baud2.SelectedIndex = 0
        End If
    End Sub

```

```

    Private Sub radioButton20_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub

```

```

    Private Sub button_OffsetTime_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
button_OffsetTime.Click
        Dim OffsetTime As Byte

```

```

        OffsetTime =
Convert.ToByte(comboBox_OffsetTime.SelectedIndex)
        fCmdRet = StaticClassReaderB.SetOffsetTime(fComAdr,
OffsetTime, frmcomportindex)
        AddCmdLog("SetOffsetTime", "SetOffsetTime", fCmdRet)
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button_BlockWrite.Click
        Dim WordPtr, ENum1 As Byte
        Dim Num As Byte = 0
        Dim Mem As Byte = 0
        Dim WNum As Byte = 0
        Dim EPCLength As Byte = 0
        Dim Writedatalen As Byte = 0
        Dim WrittenDataNum As Integer = 0
        Dim s2, str As String
        Dim CardData(320) As Byte
        Dim writedata(230) As Byte
        If ((maskadr_textbox.Text = "") Or (maskLen_textBox.Text =
"")) Then
            Exit Sub
        End If
        If (checkBox1.Checked) Then
            MaskFlag = 1
        Else
            MaskFlag = 0
        End If
        Maskadr = Convert.ToByte(maskadr_textbox.Text, 16)
        MaskLen = Convert.ToByte(maskLen_textBox.Text, 16)
        If (ComboBox_EPC2.Items.Count = 0) Then
            Exit Sub
        End If
        If (ComboBox_EPC2.SelectedItem = Nothing) Then
            Exit Sub
        End If
        str = ComboBox_EPC2.SelectedItem.ToString()
        ENum1 = Convert.ToByte(str.Length / 4)
        EPCLength = Convert.ToByte(ENum1 * 2)

```

```

Dim EPC(ENum1) As Byte
EPC = HexStringToByteArray(str)
If (C_Reserve.Checked) Then
    Mem = 0
End If
If (C_EPC.Checked) Then
    Mem = 1
End If
If (C_TID.Checked) Then
    Mem = 2
End If
If (C_User.Checked) Then
    Mem = 3
End If
If (Edit_WordPtr.Text = "") Then
    MessageBox.Show("Address of Tag Data is NULL",
"Information")
    Exit Sub
End If
If (textBox1.Text = "") Then
    MessageBox.Show("Length of Data(Read/Block Erase) is
NULL", "Information")
    Exit Sub
End If
If (Convert.ToInt32(Edit_WordPtr.Text, 16) +
Convert.ToInt32(textBox1.Text) > 120) Then
    Exit Sub
End If
If (Edit_AccessCode2.Text = "") Then
    Exit Sub
End If
WordPtr = Convert.ToByte(Edit_WordPtr.Text, 16)
Num = Convert.ToByte(textBox1.Text)
If (Edit_AccessCode2.Text.Length <> 8) Then
    Exit Sub
End If
fPassWord =
HexStringToByteArray(Edit_AccessCode2.Text)
If (Edit_WriteData.Text = "") Then

```

```

Exit Sub
End If
s2 = Edit_WriteData.Text
If (s2.Length Mod 4 <> 0) Then
    MessageBox.Show("The Number must be 4 times.",
"WriteBlock")
Exit Sub
End If
WNum = Convert.ToByte(s2.Length / 4)
ReDim writedata(WNum * 2)
writedata = HexStringToByteArray(s2)
Writedatalen = Convert.ToByte(WNum * 2)
If ((checkBox_pc.Checked) And (C_EPC.Checked)) Then
    WordPtr = 1
    Writedatalen =
Convert.ToByte(Edit_WriteData.Text.Length / 2 + 2)
    writedata = HexStringToByteArray(textBox_pc.Text +
Edit_WriteData.Text)
End If
fCmdRet = StaticClassReaderB.WriteBlock_G2(fComAdr,
EPC, Mem, WordPtr, Writedatalen, writedata, fPassWord, Maskadr,
MaskLen, MaskFlag, WrittenDataNum, EPClength, ferrorcode,
frmcomportindex)
AddCmdLog("Write data", "write", fCmdRet)
If (fCmdRet = 0) Then
    StatusBar1.Panels(0).Text =
DateTime.Now.ToLongTimeString() + ";@'WriteBlock'Command
Response=0x00'(completely Write Block successfully)"
End If
End Sub

Private Sub button13_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button13.Click
    Dim dminfre, dmaxfre, Ffenpin As Byte
    Dim i, j, CardNum, Totalen, UID_index, n_index As Integer
    Dim EPC(5000) As Byte
    Dim temp1, temp2, temp3, temp4 As String
    Dim ncount As Double
    Dim AdrTID As Byte = 0

```

```

Dim LenTID As Byte = 0
Dim TIDFlag As Byte = 0
Dim band As Byte = 0
Dim m_max As Integer = 0
button13.Enabled = False
button16.Enabled = True
button18.Enabled = False
button19.Enabled = False
listBox4.Items.Clear()
temp1 = ""
temp2 = ""
temp3 = ""
temp4 = ""
breakflag = False
If (radioButton_band1.Checked) Then
    m_max = 61
    band = 0
End If
If (radioButton_band2.Checked) Then
    m_max = 18
    band = 1
End If
If (radioButton_band3.Checked) Then
    m_max = 48
    band = 2
End If
If (radioButton_band4.Checked) Then
    m_max = 30
    band = 3
End If
If (radioButton_band5.Checked) Then
    m_max = 13
    band = 4
End If
For Ffenpin = 0 To m_max
    If (breakflag = True) Then
        breakflag = False
    If (fAppClosed) Then
        Close()
    
```

```

        End If
        Exit Sub
    End If
    dminfre = Convert.ToByte(((band And 3) << 6) Or (Ffenpin
And &H3F))
    dmaxfre = Convert.ToByte(((band And &HC) << 4) Or
(Ffenpin And &H3F))
    If (radioButton_band1.Checked) Then
        y_f = Convert.ToDouble(902.6 + (Ffenpin And &H3F) *
0.4)
    End If
    If (radioButton_band2.Checked) Then
        y_f = Convert.ToDouble(920.125 + (Ffenpin And &H3F)
* 0.25)
    End If
    If (radioButton_band3.Checked) Then
        y_f = Convert.ToDouble(902.75 + (Ffenpin And &H3F)
* 0.5)
    End If
    If (radioButton_band4.Checked) Then
        y_f = Convert.ToDouble(917.1 + (Ffenpin And &H3F) *
0.2)
    End If
    If (radioButton_band5.Checked) Then
        y_f = Convert.ToDouble(865.1 + (Ffenpin And &H3F) *
0.2)
    End If
    temp4 = Convert.ToString(y_f)
    temp3 = temp4.PadRight(5, " ") + "MHz" + "(" +
Convert.ToString(Ffenpin).PadLeft(2, " ") + ")"
    listBox4.Items.Add(temp3)
    For i = 0 To 3
        fCmdRet = StaticClassReaderB.Writedfre(fComAdr,
dmaxfre, dminfre, frmcomportindex)
        If (fCmdRet = 0) Then
            Exit For
        End If
    Next i
    ncount = 0

```

```

For j = 0 To 29
    Application.DoEvents()
    If (breakflag) Then
        breakflag = False
        If (fAppClosed) Then
            Close()
        End If
    Exit Sub
End If
CardNum = 0
Totallen = 0
fCmdRet = StaticClassReaderB.Inventory_G2(fComAdr,
AdrTID, LenTID, TIDFlag, EPC, Totallen, CardNum, frmcomportindex)
If ((fCmdRet = 1) Or (fCmdRet = 2) Or (fCmdRet = 3) Or
(fCmdRet = 4)) Then
    ncount = ncount + 1
    If (ncount = 1) Then
        UID_index = listBox4.Items.IndexOf(temp3)
    Else
        UID_index = listBox4.Items.IndexOf(temp3 + "
" + Convert.ToString(ncount - 1).PadLeft(2, " ") + "/30")
    End If
    If (UID_index >= 0) Then
        listBox4.Items(UID_index) = temp3 + "
"
+ Convert.ToString(ncount).PadLeft(2, " ") + "/30"
    End If
End If
Next j
If (ncount = 0) Then
    UID_index = listBox4.Items.IndexOf(temp3)
    If (UID_index >= 0) Then
        listBox4.Items(UID_index) = temp3 + "
"
+ Convert.ToString(ncount).PadLeft(2, " ") + "/30" + "
"
+ "00.00%"
    End If
End If
UID_index = listBox4.Items.IndexOf(temp3 + "
" + Convert.ToString(ncount).PadLeft(2, " ") + "/30")
If (UID_index >= 0) Then

```

```

        x_z = ((ncount / 30) * 100)
        temp1 = Convert.ToString(x_z)
        If (ncount = 30) Then
            temp2 = "100.00%"
        End If
        ' Else
        n_index = temp1.IndexOf(".")
        If (n_index > 0) Then
            temp2 = temp1.Substring(0, n_index) + "." +
temp1.Substring(n_index + 1, 2) + "%"
        Else
            temp2 = temp1 + "." + "00" + "%"
        End If
        listBox4.Items(UID_index) = temp3 + "
+ Convert.ToString(ncount).PadLeft(2, " ") + "/30" + "
+ temp2
    End If
    listBox4.SelectedIndex = listBox4.Items.Count - 1
Next Ffenpin
button13.Enabled = True
button16.Enabled = False
button18.Enabled = True
button19.Enabled = True
End Sub

Private Sub button16_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button16.Click
    breakflag = True
    button13.Enabled = True
    button16.Enabled = False
    button18.Enabled = True
    button19.Enabled = True
End Sub

Private Sub tabControl1_Selecting_1(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.TabControlCancelEventArgs) Handles
tabControl1.Selecting
    timer1.Enabled = False

```


button10.Text = "Get"

Timer_G2_Alarm.Enabled = False

Timer_G2_Read.Enabled = False

Timer_Test_.Enabled = False

SpeedButton_Read_G2.Text = "Read"

button2.Text = "Query Tag"

button4.Text = "Check Alarm"

If ((ListView1_EPC.Items.Count <> 0) And (ComOpen))

Then

button2.Enabled = True

DestroyCode.Enabled = False

AccessCode.Enabled = False

NoProect.Enabled = False

Proect.Enabled = False

Always.Enabled = False

AlwaysNot.Enabled = False

NoProect2.Enabled = True

Proect2.Enabled = True

Always2.Enabled = True

AlwaysNot2.Enabled = True

P_Reserve.Enabled = True

P_EPC.Enabled = True

P_TID.Enabled = True

P_User.Enabled = True

Button_DestroyCard.Enabled = True

Button_SetReadProtect_G2.Enabled = True

Button_SetEASAlarm_G2.Enabled = True

Alarm_G2.Enabled = True

NoAlarm_G2.Enabled = True

Button_LockUserBlock_G2.Enabled = True

Button_WriteEPC_G2.Enabled = True

Button_SetMultiReadProtect_G2.Enabled = True

Button_RemoveReadProtect_G2.Enabled = True

Button_CheckReadProtected_G2.Enabled = True

button4.Enabled = True

SpeedButton_Read_G2.Enabled = True

Button_SetProtectState.Enabled = True

Button_DataWrite.Enabled = True

```

Button_BlockWrite.Enabled = True
Button_BlockErase.Enabled = True
checkBox1.Enabled = True
End If
If ((ListView1_EPC.Items.Count = 0) And (ComOpen)) Then
    button2.Enabled = True
    DestroyCode.Enabled = False
    AccessCode.Enabled = False
    NoProect.Enabled = False
    Proect.Enabled = False
    Always.Enabled = False
    AlwaysNot.Enabled = False
    NoProect2.Enabled = False
    Proect2.Enabled = False
    Always2.Enabled = False
    AlwaysNot2.Enabled = False
    P_Reserve.Enabled = False
    P_EPC.Enabled = False
    P_TID.Enabled = False
    P_User.Enabled = False
    Button_DestroyCard.Enabled = False
    Button_SetReadProtect_G2.Enabled = False
    Button_SetEASAlarm_G2.Enabled = False
    Alarm_G2.Enabled = False
    NoAlarm_G2.Enabled = False
    Button_LockUserBlock_G2.Enabled = False
    SpeedButton_Read_G2.Enabled = False
    Button_DataWrite.Enabled = False
    Button_BlockErase.Enabled = False
    Button_BlockWrite.Enabled = False
    Button_WriteEPC_G2.Enabled = True
    Button_SetMultiReadProtect_G2.Enabled = True
    Button_RemoveReadProtect_G2.Enabled = True
    Button_CheckReadProtected_G2.Enabled = True
    button4.Enabled = True
    Button_SetProtectState.Enabled = False
    checkBox1.Enabled = False
End If

```

```

Timer_Test_6B.Enabled = False
Timer_6B_Read.Enabled = False
Timer_6B_Write.Enabled = False
SpeedButton_Query_6B.Text = "Query"
SpeedButton_Read_6B.Text = "Read"
SpeedButton_Write_6B.Text = "Write"
If ((ListView_ID_6B.Items.Count <> 0) And (ComOpen))
Then
    SpeedButton_Query_6B.Enabled = True
    SpeedButton_Read_6B.Enabled = True
    SpeedButton_Write_6B.Enabled = True
    Button14.Enabled = True
    Button15.Enabled = True
    If (Bycondition_6B.Checked) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
If ((ListView_ID_6B.Items.Count = 0) And (ComOpen)) Then
    SpeedButton_Query_6B.Enabled = True
    SpeedButton_Read_6B.Enabled = False
    SpeedButton_Write_6B.Enabled = False
    Button14.Enabled = False
    Button15.Enabled = False
    If (Bycondition_6B.Checked) Then
        Same_6B.Enabled = True
        Different_6B.Enabled = True
        Less_6B.Enabled = True
        Greater_6B.Enabled = True
    End If
End If
breakflag = True
button13.Enabled = ComOpen
button16.Enabled = False
button18.Enabled = ComOpen
button19.Enabled = ComOpen

```

End Sub

```
Private Sub button17_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles button17.Click  
    listBox4.Items.Clear()  
End Sub
```

```
Private Sub button18_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles button18.Click  
    Dim FlashMode As Byte  
    FlashMode = Convert.ToByte(comboBox8.SelectedIndex)  
    fCmdRet = StaticClassReaderB.SetFhssMode(fComAdr,  
FlashMode, frmcomportindex)  
    AddCmdLog("SetFhssMode", "SetFhssMode", fCmdRet)  
End Sub
```

```
Private Sub button19_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles button19.Click  
    Dim FlashMode As Byte  
    FlashMode = 0  
    fCmdRet = StaticClassReaderB.GetFhssMode(fComAdr,  
FlashMode, frmcomportindex)  
    If (fCmdRet = 0) Then  
        comboBox8.SelectedIndex = FlashMode  
    End If  
    AddCmdLog("GetFhssMode", "GetFhssMode", fCmdRet)  
End Sub
```

```
Private Sub radioButton_band5_CheckedChanged(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
radioButton_band5.CheckedChanged  
    Dim i As Integer  
    ComboBox_dminfre.Items.Clear()  
    ComboBox_dmaxfre.Items.Clear()  
    For i = 0 To 14  
        ComboBox_dminfre.Items.Add(Convert.ToString(865.1 + i  
* 0.2) + " MHz")  
        ComboBox_dmaxfre.Items.Add(Convert.ToString(865.1 +  
i * 0.2) + " MHz")  
    Next i
```

```

        Next i
        ComboBox_dmaxfre.SelectedIndex = 14
        ComboBox_dminfre.SelectedIndex = 0
    End Sub

    Private Sub Edit_WriteData_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Edit_WriteData.TextChanged
        Dim n As Integer = 0
        Dim m As Integer = 0
        n = Edit_WriteData.Text.Length
        If ((checkBox_pc.Checked) And (n Mod 4 = 0) And
(C_EPC.Checked)) Then
            m = n / 4
            m = (m And &H3F) << 3
            textBox_pc.Text = Convert.ToString(m, 16).PadLeft(2, "0")
+ "00"
        End If
    End Sub

    Private Sub checkBox_pc_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
checkBox_pc.CheckedChanged
        Dim n As Integer = 0
        Dim m As Integer = 0
        If (checkBox_pc.Checked) Then
            Edit_WordPtr.Text = "02"
            Edit_WordPtr.ReadOnly = True
            n = Edit_WriteData.Text.Length
            If ((checkBox_pc.Checked) And (n Mod 4 = 0) And
(C_EPC.Checked)) Then
                m = n / 4
                m = (m And &H3F) << 3
                textBox_pc.Text = Convert.ToString(m, 16).PadLeft(2,
"0") + "00"
            End If
        Else
            Edit_WordPtr.ReadOnly = False
        End If
    End Sub

```

End Sub

```
Private Sub CheckBox_TID_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox_TID.CheckedChanged
    If (CheckBox_TID.Checked) Then
        groupBox33.Enabled = True
        textBox4.Enabled = True
        textBox4.Enabled = True
    Else
        groupBox33.Enabled = False
        textBox4.Enabled = False
        textBox4.Enabled = False
    End If
End Sub
```

```
Private Sub button_settigtime_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
button_settigtime.Click
    Dim TriggerTime As Byte
    TriggerTime =
Convert.ToByte(comboBox_tigtime.SelectedIndex)
    fCmdRet = StaticClassReaderB.SetTriggerTime(fComAdr,
TriggerTime, frmcomportindex)
    AddCmdLog("SetTriggerTime", "Set TriggerTime",
fCmdRet)
End Sub
```

```
Private Sub button_gettigtime_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
button_gettigtime.Click
    Dim TriggerTime As Byte
    TriggerTime = 255
    fCmdRet = StaticClassReaderB.SetTriggerTime(fComAdr,
TriggerTime, frmcomportindex)
    If (fCmdRet = 0) Then
        comboBox_tigtime.SelectedIndex = TriggerTime
    End If
```

```

        AddCmdLog("SetTriggerTime", "Get TriggerTime",
fCmdRet)
    End Sub

```

```

    Private Sub button20_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button20.Click
        Dim RelayStatus As Byte = 0
        If (comboBox9.SelectedIndex = 0) Then
            RelayStatus = Convert.ToByte(RelayStatus Or 0)
        Else
            RelayStatus = Convert.ToByte(RelayStatus Or 1)
        End If
        If (comboBox10.SelectedIndex = 0) Then
            RelayStatus = Convert.ToByte(RelayStatus Or 0)
        Else
            RelayStatus = Convert.ToByte(RelayStatus Or 2)
        End If
        fCmdRet = StaticClassReaderB.SetRelay(fComAdr,
RelayStatus, frmcomportindex)
        AddCmdLog("SetRelay", "Set", fCmdRet)
    End Sub

```

```

    Private Sub button21_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button21.Click
        listView1.Items.Clear()
        Dim str As String = ""
        Dim i As Integer = 0
        Dim temps As String = ""
        IP = ""
        username = ""
        dsname = ""
        mac = ""
        portnum = ""
        tup = ""
        rm = ""
        cm = ""
        ct = ""
        fc = ""
        dt = ""

```

```

br = ""
pr = ""
bb = ""
rc = ""
ml = ""
md = ""
di = ""
dp = ""
gi = ""
nm = ""
Try
    Dim aListItem As ListViewItem
    Dim sock As New Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp)
    Dim broadcast As IPAddress = IPAddress.Broadcast
    'Dim broadcast As IPAddress =
IPAddress.Parse("192.168.1.30")
    Dim request As String = "X"
    Dim data As Byte() = Encoding.ASCII.GetBytes(request)
    Dim ep As New IPEndPoint(broadcast, 65535)
    sock.SetSocketOption(SocketOptionLevel.Socket,
SocketOptionName.Broadcast, 1)
    sock.SendTo(data, ep)
    Dim buffer(300) As Byte
    Dim m_count As Integer = 0
    sock.ReceiveTimeout = 10
    For i = 0 To 254
        m_count = sock.ReceiveFrom(buffer, ep)
        If (m_count > 0) Then
            aListItem =
listView1.Items.Add((listView1.Items.Count + 1).ToString())
            aListItem.SubItems.Add("")
            aListItem.SubItems.Add("")
            aListItem.SubItems.Add("")
            Dim daw(m_count) As Byte
            Array.Copy(buffer, daw, m_count)
            Dim fRecvIDPstring As String = ""
            fRecvIDPstring = Encoding.ASCII.GetString(daw)
            If (fRecvIDPstring.Substring(0, 1) = "A") Then

```



```

    Dim m As Integer
    m = fRecvIDPstring.IndexOf("/")
    mac = fRecvIDPstring.Substring(1, m - 1)
    portnum = fRecvIDPstring.Substring(m + 1, 4)
    m = fRecvIDPstring.IndexOf("*:")
    Dim n As Integer
    n = fRecvIDPstring.Length - m - 8
    Dim IDPstring As String = ""
    IDPstring = fRecvIDPstring.Substring(m + 8, n)
    str = ""
    If (IDPstring <> "") Then
        username = IDPstring.Substring(0,
IDPstring.IndexOf("/"))
        dsname =
IDPstring.Substring(IDPstring.IndexOf("/") + 1, IDPstring.Length -
IDPstring.IndexOf("/") - 1)
        IP = ep.ToString().Substring(0,
ep.ToString().IndexOf(":"))
        If (((username = "") And (dsname = "")) Or
(dsname = "/")) Then
            str = ""
        Else
            str = username + "/" + dsname
        End If
    End If
    aListItem.SubItems(1).Text = mac
    aListItem.SubItems(2).Text = IP
    aListItem.SubItems(3).Text = str
End If
End If
Next
sock.Close()
Catch ex As Exception
    ex.ToString()
Exit Sub
End Try
End Sub

```

```

        Private Sub button26_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles button26.Click
            Dim i As Integer = 0
            Try
                If ((listView1.SelectedIndices.Count > 0) And
(listView1.SelectedIndices(0) <> -1)) Then
                    IP = listView1.SelectedItems(0).SubItems(2).Text
                    Dim sock As New Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp)
                    Dim broadcast As IPAddress = IPAddress.Parse(IP)
                    Dim ep As New IPEndPoint(broadcast, 65535)
                    'EndPoint ep = (EndPoint)iep;
                    sock.SetSocketOption(SocketOptionLevel.Socket,
SocketOptionName.Broadcast, 1)
                    Dim request As String = "B"
                    Dim buffer() As Byte
                    buffer = Encoding.ASCII.GetBytes(request)
                    While (i < 3)
                        sock.SendTo(buffer, ep)
                        Dim buffer1(1000) As Byte
                        sock.ReceiveTimeout = 10
                        Dim m_count As Integer
                        m_count = sock.ReceiveFrom(buffer1, ep)
                        If (m_count > 0) Then
                            i = 3
                            sock.Close()
                            Exit While
                        End If
                        i = i + 1
                    End While
                Else
                    MessageBox.Show("No select device!", "Information")
                End If
            Catch ex As Exception
                ex.ToString()
            End Try
        End Sub

```

Lampiran 2. Hasil Pengujian Pemindaian Peralatan (30ms)

No	Antenna Power	Inventory Scantime	Read interval	Peralatan dalam lemari	Peralatan yang terdeteksi
1	15	20*100ms	30ms	12	12
2	15	20*100ms	30ms	12	12
3	15	20*100ms	30ms	12	12
4	15	20*100ms	30ms	12	12
5	15	20*100ms	30ms	12	12
6	15	20*100ms	30ms	12	12
7	15	20*100ms	30ms	12	11
8	15	20*100ms	30ms	12	12
9	15	20*100ms	30ms	12	10
10	15	20*100ms	30ms	12	12
11	15	20*100ms	30ms	12	12
12	15	20*100ms	30ms	12	12
13	15	20*100ms	30ms	12	12
14	15	20*100ms	30ms	12	12
15	15	20*100ms	30ms	12	12
16	15	20*100ms	30ms	12	12
17	15	20*100ms	30ms	12	12
18	15	20*100ms	30ms	12	12
19	15	20*100ms	30ms	12	12
20	15	20*100ms	30ms	12	12
21	15	20*100ms	30ms	12	11
22	15	20*100ms	30ms	12	12
23	15	20*100ms	30ms	12	12
24	15	20*100ms	30ms	12	12
25	15	20*100ms	30ms	12	12

26	15	20*100ms	30ms	12	12
27	15	20*100ms	30ms	12	12
28	15	20*100ms	30ms	12	12
29	15	20*100ms	30ms	12	12
30	15	20*100ms	30ms	12	10
31	15	20*100ms	30ms	12	12
32	15	20*100ms	30ms	12	12
33	15	20*100ms	30ms	12	12
34	15	20*100ms	30ms	12	11
35	15	20*100ms	30ms	12	12
36	15	20*100ms	30ms	12	12
37	15	20*100ms	30ms	12	12
38	15	20*100ms	30ms	12	10
39	15	20*100ms	30ms	12	12
40	15	20*100ms	30ms	12	12
41	15	20*100ms	30ms	12	12
42	15	20*100ms	30ms	12	11
43	15	20*100ms	30ms	12	12
44	15	20*100ms	30ms	12	12
45	15	20*100ms	30ms	12	12
46	15	20*100ms	30ms	12	12
47	15	20*100ms	30ms	12	12
48	15	20*100ms	30ms	12	12
49	15	20*100ms	30ms	12	12
50	15	20*100ms	30ms	12	12

Lampiran 3. Hasil Pengujian Pemindaian Peralatan (50ms)

No	Antenna Power	Inventory Scantime	Read interval	Peralatan dalam lemari	Peralatan yang terdeteksi
1	15	20*100ms	50ms	12	10
2	15	20*100ms	50ms	12	12
3	15	20*100ms	50ms	12	12
4	15	20*100ms	50ms	12	12
5	15	20*100ms	50ms	12	10
6	15	20*100ms	50ms	12	12
7	15	20*100ms	50ms	12	12
8	15	20*100ms	50ms	12	12
9	15	20*100ms	50ms	12	11
10	15	20*100ms	50ms	12	12
11	15	20*100ms	50ms	12	10
12	15	20*100ms	50ms	12	12
13	15	20*100ms	50ms	12	11
14	15	20*100ms	50ms	12	12
15	15	20*100ms	50ms	12	12
16	15	20*100ms	50ms	12	12
17	15	20*100ms	50ms	12	12
18	15	20*100ms	50ms	12	12
19	15	20*100ms	50ms	12	12
20	15	20*100ms	50ms	12	8
21	15	20*100ms	50ms	12	12
22	15	20*100ms	50ms	12	12
23	15	20*100ms	50ms	12	4
24	15	20*100ms	50ms	12	12
25	15	20*100ms	50ms	12	11

26	15	20*100ms	50ms	12	10
27	15	20*100ms	50ms	12	12
28	15	20*100ms	50ms	12	11
29	15	20*100ms	50ms	12	12
30	15	20*100ms	50ms	12	12
31	15	20*100ms	50ms	12	10
32	15	20*100ms	50ms	12	12
33	15	20*100ms	50ms	12	12
34	15	20*100ms	50ms	12	12
35	15	20*100ms	50ms	12	12
36	15	20*100ms	50ms	12	9
37	15	20*100ms	50ms	12	12
38	15	20*100ms	50ms	12	10
39	15	20*100ms	50ms	12	12
40	15	20*100ms	50ms	12	8
41	15	20*100ms	50ms	12	12
42	15	20*100ms	50ms	12	12
43	15	20*100ms	50ms	12	8
44	15	20*100ms	50ms	12	12
45	15	20*100ms	50ms	12	11
46	15	20*100ms	50ms	12	12
47	15	20*100ms	50ms	12	12
48	15	20*100ms	50ms	12	7
49	15	20*100ms	50ms	12	12
50	15	20*100ms	50ms	12	12

Lampiran 4. Manual Book *Automation Tool's Crib*

1 KESELAMATAN

A. PERINGATAN KESELAMATAN

1. PERINGATAN UMUM

Mesin ini dilengkapi dengan alat keselamatan untuk melindungi pengguna dan mesin dari bahaya kecelakaan. Operasi yang tidak sesuai prosedur dapat mengakibatkan kecelakaan. Oleh sebab itu, operator harus membaca buku petunjuk penggunaan mesin sebelum mengoperasikan dan melakukan perawatan mesin. Apabila ada fungsi yang tidak dijelaskan dalam buku ini, silahkan menghubungi Departemen Technical Service.

Sebelum mengoperasikan mesin, disarankan untuk membaca peringatan umum untuk mencegah kecelakaan.

1. Mesin ini harus dioperasikan oleh personil **terlatih**.
2. Disarankan untuk tidak mengambil *Tools* lebih dari 3
3. Operasikanlah mesin ini dalam keadaan **sehat** dan **fit** untuk bekerja.
4. Area disekitar bagian operasi mesin harus dilengkapi dengan **pencahaya**an yang **sesuai** dan perlengkapan pengaman.
5. **Lantai** pabrik harus dalam keadaan **bersih** dan **kering**.
6. **Rambut** yang **panjang** harus **diikat** dengan rapi.

2. TINDAKAN SAFETY

Gunakan Alat Pelindung Diri (APD) seperti gambar di bawah ini :



I. Safety action ketika mesin dalam keadaan running dan beroperasi

1. Pastikan tangan/ anggota badan anda dalam keadaan kering dan tidak basah.
2. Pastikan **Admin mengecek** keteraturan kinerja mesin , proses, dan hasilnya secara **berkala**.

II. Safety Action ketika mesin dalam keadaan bermasalah dan macet

1. Ketika terjadi masalah, tekan EMERGENCY.

2. Pastikan mesin berada dalam keadaan **MATI/OFF** sebelum tindakan / penanganan selanjutnya.

III. Buka Bab 4 buku bagian Maintenance/Troubleshooting

B. WARNING LABEL

1. Karena *warning label* menempel pada bagian-bagian mesin agar dapat beroperasi dengan aman, pahami isi *warning label* sebelum mengoperasikan mesin.
2. Jaga kebersihan *warning label* sehingga dapat dibaca dengan jelas.
3. Jangan pindahkan *warning label*. Jika *warning label* dipindahkan, tempelkan kembali ke posisi semula.



Mengindikasikan adanya **tegangan listrik tinggi (berbahaya)**, kegagalan dalam mengikuti instruksi akan menyebabkan kematian atau luka serius.



Instruksi untuk **menutup pintu** agar terhindar dari kemungkinan bahaya.

*CATATAN WARNING LABEL

Peringatan *Warning Label*

1. Pahami seluruh *warning label* mesin ini.
2. Jaga kebersihan *warning label* agar terbaca jelas.

3. Biarkan posisi *warning label* sesuai tempatnya.

DESKRIPSI MESIN

A. MACHINE

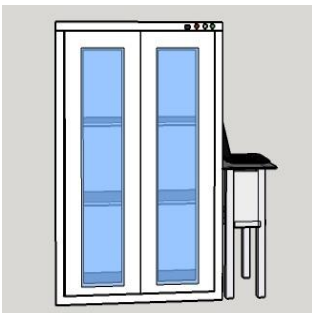


Integrasi Mesin *Tools Crib* merupakan integrasi antara tiga device yang berbeda yaitu Visual Studio, PLC dan RFID

B.

C. SPESIFIKASI & PRINSIP KERJA MESIN

Dimensi	Panjang (mm)	Lebar (mm)	Tinggi (mm)
	18,850 (total 2 line)	7,227	3,225



Urutan proses :

1. Login menggunakan kartu RFID dengan mendekatkan ke lemari
2. Setelah mendengar bunyi *Door Lock*, tunggu sebentar lalu buka pintu.
3. Ambil *Tools* sesuai dengan yang dibutuhkan (maks 3 *tools*).
4. Tutup pintu setelah mengambil *tools*.
5. Lihat pada layar HMI sudah muncul *Transaction Successfull*.

KEBUTUHAN DAYA		
Tegangan	220v	
Arus	1A	
Konsumsi daya	240W	
FITUR		
1. Transaksi otomatis tanpa operator		
2. Pembukuan otomatis dengan database		
3. Identifikasi Alat dan pengguna Otomatis dengan RFID		
4. membutuhkan transaksi < 2 menit		
5. Maintenance data dan peralatan dapat diawasi		
6. Waktu operasi 24 jam		
KAPASITAS		
lebar rak	47 cm	
panjang rak	87.5 cm	
jumlah rak	3	
kapasitas RFID	255 tools	
Kapasitas Lemari	20	Large size tools
	40	medium Size tools
DIMENSI DAN LAYOUT		
Lemari		
Tinggi	1,4 m	

Panjang	0,5
Lebar	0,5
Layout	
Tinggi	2 m
panjang	1,5 m
lebar	1 m

OPERASI & PERAWATAN

A. PERSIAPAN SEBELUM BEROPERASI

1. Pastikan tidak ada bahaya di sekitar mesin.
2. Pastikan alat keselamatan tersedia.
3. Periksa dan pastikan kabel sudah tersambung dengan power source.

B. MENYALAKAN MESIN

1. Nyalakan mesin dengan mendorong *switch* NFB dari posisi OFF ke posisi ON



NFB Master Panel

2. Nyalakan PLC pada NFB Box dengan mendorong *switch* pada posisi ON.





C. DESKRIPSI PANEL OPERASI

D. OPERASI MESIN



PENGOPERASIAN MANUAL

1. Periksa apakah kabel *power* terhubung dengan benar.
2. Aktifkan NFB utama untuk menyalakan mesin.
3. Pastikan lampu *indicator power* menyala.
4. Pastikan tidak muncul alarm ditandai dengan *indicator tower lamp* berwarna merah. Apabila *indicator tower lamp* berwarna merah, buka software *Automation Tools Crib*.
5. Pada *operation panel* pilih *Key Switch* dengan seizin admin.
6. Pada Form Admin pilih login manual
7. Masukkan NIK dan Password

PENGOPERASIAN OTOMATIS

1. Pastikan tidak ada bahaya di sekitar peralatan.
2. Pastikan peralatan keselamatan tersedia.
3. Periksa apakah kabel terhubung dengan benar.
4. Pastikan lampu *indicator* menyala.
5. Pastikan tidak muncul alarm ditandai dengan *indicator tower lamp* berwarna merah. Apabila *indicator tower lamp* berwarna merah, buka *Software Automation Tools Crib*.
6. Tempelkan kartu pada kaca lemari untuk login.
7. Buka pintu setelah terdengar bunyi *Door Lock*.

8. Ambil *Tools* sesuai kebutuhan atau kembalikan *Tools* sesuai dengan tempatnya.
9. Tutup pintu dengan rapat.
10. Proses selesai.

TROUBLESHOOTING

Apabila terjadi masalah (error) selama mesin beroperasi :

1. Tekan tombol emergency.
2. Nonaktifkan tombol dengan memutar tombol ke kiri.
3. Ikuti petunjuk troubleshooting di bawah ini :

1) HMI

Problem	Kemungkinan Penyebab	Tindakan Perbaikan
1. Program berhenti tiba – tiba	- Program Not responding	- Buka kembali program .Exe
1. <i>Object out of String</i>	- Kabel Ethernet terlepas	- Pasang kembali kabel Ethernet - Lepaskan dan masukan kebalik kabel Ethernet ke USB
2. <i>Serial Comm Error</i>	- USB RFID terlepas	- Pasang kembali kabel RFID - Lepaskan dan masukan kebalik kabel RFID ke USB

Lampiran 5. Datasheet RFID

Model No.	CT-I809
Type	UHF Integrative RFID Reader
Working frequency	902-928Mhz or 865-868Mhz
Working mode	FHSS hopping and fixed frequency
Support protocol	ISO18000-6C/EPC Gen2 ,ISO18000-6B
Transmit power	0 ~ 30dBm adjustable
Antenna mode	Built-in 8dbi polarization antenna
Read &Write distance	read 1-6m, write 1-3m, tag and environment dependent
Data Interface	TCPIP/RJ45, RS232/UART, RS485, Weigand26/34
Read prompt	Buzzer
Power supply	single +9V DC
Unit size	235mmx235mmx57mm
Net weight	900g
Operating temperature	-10 ~ +60°C
Storage temperature	-25 ~ +80°C
Humidity	5%~95%(non-condensing)
Material	ABS Waterproof



Cable Connector

Color	Type
Red	+9V
Black	GND
Yellow	Weigand DATA0
Blue	Weigand DATA1
Purple	RS485 R+
Orange	RS485 R-
Brown	GND
White	RS232 RXD
Green	RS232 TXD
Grey	Trigger input (TTL level)

Lampiran 6. Datasheet PLC CJ1M-CPU13

Item	Specifications
Control method	Stored program
I/O control method	Cyclic scan and immediate processing are both possible.
Programming	LD (Ladder), SFC (Sequential Function Chart), ST (Structured Text), Mnemonic
CPU processing mode	CJ1M CPU Units: Normal Mode or Peripheral Servicing Priority Mode
Instruction length	1 to 7 steps per instruction
Ladder instructions	Approx. 400 (3-digit function codes)
Execution time	CJ1M CPU Units (CPU12/13/22/23): Basic instructions: 0.10 μ s min. Special instructions: 0.15 μ s min. CJ1M CPU Units (CPU11/21): Basic instructions: 0.10 μ s min. Special instructions: 0.15 μ s min.
Overhead time	CJ1M CPU Units (CPU12/13/22/23): 0.5 ms min. CJ1M CPU Units (CPU11/21): 0.7 ms min.
Unit connection method	No Backplane: Units connected directly to each other.
Mounting method	DIN Track (screw mounting not possible)
Maximum number of connectable Units	CJ1M CPU Units: Total of 20 Units in the System, including 10 Units on CPU Rack and 10 Units on one Expansion Rack.
Maximum number of Expansion Racks	CJ1M CPU Units (CPU 13/23 only): 1 max. (An I/O Control Unit is required on the CPU Rack and an I/O Interface Unit is required on the Expansion Rack.) CJ1M CPU Units (CPU11/12/21/22): Expansion is not possible.

Lampiran 7. Datasheet Input Card CJ1M - ID211

Unit type	Product name	Specifications					Current consumption (A)		Model	Standards	
		I/O points	Input voltage and current	Commons	External connection	No. of words allocated	5 V	24 V			
CJ1 Basic I/O Units		DC Input Units	8 inputs	12 to 24 VDC, 10 mA	Independent contacts	Removable terminal block	1 word	0.09	–	CJ1W-ID201	UC1, N, L, CE
			16 inputs	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.08	–	CJ1W-ID211	
			16 inputs (High speed)	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.13	–	CJ1W-ID212	N, L, CE
			32 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu connector	2 words	0.09	–	CJ1W-ID231	
			32 inputs	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	2 words	0.09	–	CJ1W-ID232	UC1, N, L, CE
			32 inputs (High speed)	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	2 words	0.20	–	CJ1W-ID233	
			64 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu connector	4 words	0.09	–	CJ1W-ID261	
			64 inputs	24 VDC, 4.1 mA	16 points, 1 common	MIL connector	4 words	0.09	–	CJ1W-ID262	
		AC Input Units	8 inputs	200 to 24 VAC, 10 mA (200 V, 50 Hz)	8 points, 1 common	Removable Terminal Block	1 words	0.08	–	CJ1W-IA201	UC1, N, L, CE
			16 inputs	100 to 120 VAC, 7 mA (100 V, 50 Hz)	16 points, 1 common	Removable Terminal Block	1 words	0.09	–	CJ1W-IA111	

8.27 x 11.69 in

iii

Lampiran 8. Datasheet CJ1M – PA202

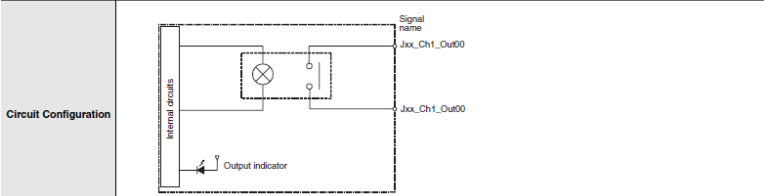
Specifications					
Item					
Model	CJ1W-PA205R	CJ1W-PA205C	CJ1W-PA202	CJ1W-PD025	CJ1W-PD022
Supply voltage	100 to 240 V AC (wide-range), 50/60 Hz			24 VDC	
Operating voltage and frequency ranges	85 to 264 V AC, 47 to 63 Hz			19.2 to 28.8 V DC	21.6 to 26.4 V DC
Power consumption	100 VA max.		50 VA max.	50 W max.	35 W max.
Inrush current *1	At 100 to 120 V AC: 15 A/8 ms max. for cold start at room temperature At 200 to 240 V AC: 30 A/8 ms max. for cold start at room temperature		At 100 to 120 V AC: 20 A/8 ms max. for cold start at room temperature At 200 to 240 V AC: 40 A/8 ms max. for cold start at room temperature	At 24 V DC: 30 A/20 ms max. for cold start	
Output capacity *7	5.0 A, 5 V DC (including supply to CPU Unit)		2.8 A, 5 V DC (including supply to CPU Unit)	5.0 A, 6 V DC (including supply to CPU Unit)	2.0 A, 5 V DC (including supply to CPU Unit)
	0.8 A, 24 V DC		0.4 A, 24 V DC	0.8 A, 24 V DC	0.4 A, 24 V DC
	Total: 25 W max.		Total: 14 W max.	Total: 25 W max.	Total: 19.6 W max.
Output terminal (service supply)	Not provided.				
RUN output *2	Contact configuration: SPST-NO Switch capacity: 250 V AC, 2 A (resistive load) 120 V AC, 0.5 A (inductive load), 24 V DC, 2 A (resistive load) 24 V DC, 2 A (inductive load)	Not provided.			
Replacement notification function	Not provided.	With Alarm output (open-collector output) 30 V DC max., 50 mA max.	Not provided.		
Insulation resistance	20 MΩ min. (at 500 V DC) between AC external and GR terminals *3	• 20 MΩ min. (at 500 V DC) between all external terminals and GR terminal *3, and between all alarm output terminals. • 20 MΩ 1 min. (at 250 V DC) between all alarm output terminals and GR terminal *3.		20 MΩ min. (at 500 V DC) between AC external and GR terminals *3	20 MΩ min. (at 500 V DC) between DC external and GR terminals *3
		* 2,300 VAC, 50/60 Hz for			– *6

Lampiran 9. Datasheet Output Card CJ1M – OD211

CJ1W-OC/OA/OD

Specifications

CJ1W-OC201 Contact Output Unit (Independent Relays, 8 Points)	
Name	8-point Contact Output Unit with Terminal Block (Independent Relays)
Model	CJ1W-OC201
Max. Switching Capacity	2 A 250 VAC (cosφ = 1), 2 A 250 VAC (cosφ = 0.4), 2 A 24 VDC (16 A/Unit)
Min. Switching Capacity	1 mA 5 VDC
Relays	NY-24W-K-IE (Fujitsu Takamizawa Components, Ltd.). Cannot be replaced.
Service Life of Relay	Electrical: 150,000 operations (24 VDC, resistive load)/100,000 operations (240 VAC, cosφ = 0.4, inductive load) Mechanical: 20,000,000 operations Service life will vary depending on the connected load.
ON Response Time	15 ms max.
OFF Response Time	15 ms max.
Number of Circuits	8 independent contacts
Insulation Resistance	20 MΩ between external terminals and the GR terminal (500 VDC)
Dielectric Strength	2,000 VAC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	90 mA 5 VDC max. 48 mA 24 VDC max. (6 mA × No. of ON points)
Weight	140 g max.



Lampiran 10. Datasheet Ethernet Card CJ1M-ETN13

CPU Functional Element

Item		Specification		
Model		CJ1M-CPU13-ETN	CJ1M-CPU12-ETN	CJ1M-CPU11-ETN
Functional element version		Ver. 3.0 or later		
I/O points		640	320	160
User program memory		20 Ksteps	10 Ksteps	5 Ksteps
Maximum number of Expansion Racks		1 max.	Not supported.	
CJ-series Basic I/O Units CJ-series Special I/O Units CJ-series CPU Bus Units		Total: 9 + 10 Units max. (CPU Rack: 9 Units Expansion Rack: 10 Units)	Total: 9 Units max.	
Data Memory		32 Kwords		
Extended Data Memory		Not supported.		
Pulse functions		Not supported.		
Interrupt Inputs		2		1
PWM output points		None		
Maximum subroutine number		1,024		256
Maximum jump number for JMP instruction		1,024		256
Function blocks (FB)	Maximum number of definitions	128		
	Maximum number of Instances	256		
Flash memory	Function block program memory	256 Kbytes		
	Comment file	64 Kbytes		
	Program Index file	64 Kbytes		
	Symbol tables	64 Kbytes		

Lampiran 11. Datasheet LED indicator

■ Lamp ratings
▪ Illuminated pushbuttons, illuminated selectors, pilot lights

Transformer	Lamp voltage	LED			Incandescent		
		Type	Rated voltage	Consumption	Type	Rated voltage	Consumption
Without transformer	5.5V AC/DC	—	—	—	AHX135	6.3V AC/DC	0.9W
	6V AC	APX510-6□	6V AC	Green, red, orange, amber, blue: 7mA AC Yellow: 50mA AC	—	—	—
	6V DC	APX510-D6□	6V DC	Green, red, orange, amber, blue: 11mA DC Yellow: 33mA DC	—	—	—
	12V AC/DC	APX510-12□	12V AC/DC	Green, red, orange, amber, blue: 14mA AC, 11mA DC Yellow: 28mA AC, 22mA DC	—	—	—
	15V AC/DC	APX510-15□	15V AC/DC	Green, red, orange, amber, blue: 13mA AC, 11mA DC Yellow: 26mA AC, 22mA DC	AHX279	18V AC/DC	0.8W
	20V AC/DC 24V AC/DC	— APX510-24□	— 24V AC/DC	— 12mA AC, 11mA DC	AHX144 AHX129	24V AC/DC 30V AC/DC	0.9W 0.8W
With transformer (Standard type: AR9T511)	110V AC	APX510-6□	6V AC	1.5VA	AHX135	6.3V AC/DC	2VA
	127V AC						2VA
	220V AC						2VA
	254V AC	APX510-6□	6V AC	2.5VA	AHX135	6.3V AC/DC	2.5VA
	380V AC						2.5VA
	440V AC 480V AC 550V AC						2.5VA 2.5VA 2.5VA
With resistor unit (AR9T519-H)	110V DC	APX510-24□	24V AC/DC	1.2W	—	—	—

Lampiran 12. Datasheet RFID TAG

SPECIFICATIONS	
	ISO Card UHF or UHF / MIFARE
Base Model Number	6A7401 (UHF), 6C3401 (UHF + MIFARE DESFire EV1 4K) 6C2401 (UHF + MIFARE Classic EV1 1K)
Dimensions	ISO card: 3.4 x 2.1 x 0.03 in (85.6 x 54 x 0.76 mm)
Material	PVC (default, others on request)
Color	White
Chip Type	Monza 4QT + (optional MIFARE DESFire EV1 4K or MIFARE EV1 1K)
Memory	UHF: 128 bit EPC, 512 bit user memory, 96 bit TID MIFARE DESFire: 4KB (optional: 256B, 2KB, 8KB), MIFARE EV1 1KB (optional: 4KB)
Operating Frequency	860-960 MHz (Worldwide) + 13.56 MHz (MIFARE)
Standards	UHF: EPC Class 1 Gen 2, ISO 18000-6C MIFARE DESFire: ISO 14443 / NFC Tag Type 4 (if NDEF formatted); ISO 10373, ISO 7816-1
Anti Collision	Yes

CHEMICAL AND MECHANICAL RESISTANCE	
Water IP68	68° F / 20° C (1m, 24h)
Resistances	Acetic acid water, artificial perspiration, carbonated sodium water, ethylene glycol, fuel B, salt mist, salt water, sugared water; humidity 95% at 122° F (50° C) 24 h
Storage Temperature	-31° to +122° F (-35° to +50° C)
Operating Temperature	-31° to +122° F (-35° to +50° C)
Shock / Fatigue Temperature	-31° to +176° F (-35° to +80° C), 50 cycles, 5 min soaking time, 20 sec transition
Force	Dynamic bending and torsion, 4 x 250 bends
Reading Distance	UHF up to 39 ft / 12 m (2W reader ERP, free space)
Options	Artwork; encoding; laser engraving; magnetic stripe; other contactless-, contact- or memory-chips; signature panel and optical security features ; UHF only cards support an optional horizontal punch slot
Prelaminates	For card manufacturers this technology is alternatively available in form of 410 µm (16 mil) prelamines

RIWAYAT PENULIS



Aldo Rahmad Ageng lahir di Surabaya tanggal 29 Oktober 1997. Penulis merupakan anak pertama dari dua bersaudara. Setelah kewajiban 12 tahun belajar dari Sekolah Dasar hingga Sekolah Menengah Atas di Surabaya. Penulis melanjutkan ke jenjang perguruan tinggi di Jurusan D3 Teknik Elektro Institut Teknologi Sepuluh Nopember, Surabaya. Penulis mempunyai kemauan untuk berstrategi dan terencana alur dari perkuliahannya serta memiliki kegigihan untuk terus belajar.

Email : Aldorahmada@gmail.com